

UNIVERSITÉ DE MONTRÉAL

CONCEPTION D'UN SYSTÈME DE VISION PAR ORDINATEUR POUR LA
DéTECTION AUTOMATIQUE D'INVENTAIRE MÉDICAL

FRÉDÉRIC BLOUIN
DÉPARTEMENT DE GÉNIE INFORMATIQUE ET GÉNIE LOGICIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE INFORMATIQUE)
AOÛT 2014

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

CONCEPTION D'UN SYSTÈME DE VISION PAR ORDINATEUR POUR LA
DÉTECTION AUTOMATIQUE D'INVENTAIRE MÉDICAL

présenté par : BLOUIN Frédéric

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de :

M. GAGNON Michel, Ph.D., président

M. LANGLOIS J.M. Pierre, Ph.D., membre et directeur de recherche

M. BILODEAU Guillaume-Alexandre, Ph.D., membre et codirecteur de recherche

Mme CHERIET Farida, Ph.D., membre

REMERCIEMENTS

Je tiens à remercier ici quelques personnes importantes pour moi, sans qui ce projet n'aurait pas eu lieu. D'abord, mes parents et mes amis proches, pour m'avoir encouragé durant toutes mes années d'études. Ensuite, mes deux directeurs de recherche, qui ont su me guider et me redonner confiance en mon travail. Je voudrais aussi remercier aussi mes collègues de travail, Diana Gil et Pierre Langlois, qui ont beaucoup contribué à ce projet, en travaillant chez Logi-D avec moi.

Finalement, je remercie tous les organismes qui ont rendu ce projet réalisable. Je pense à la compagnie Logi-D, qui m'a permis de vivre une expérience agréable de travail. Je pense aussi à l'organisme MITACS qui, grâce à son programme de bourse MITACS-Accélération, a pu m'apporter un soutien financier. Enfin, je remercie l'Hôpital du Sacré-Cœur de Montréal, ainsi que le personnel de la clinique de chirurgie d'un jour, d'avoir accepté l'installation de notre projet pilote.

RÉSUMÉ

La gestion d’inventaire dans les hôpitaux constitue un problème de logistique important. Le personnel des hôpitaux dépend quotidiennement de l’utilisation de centaines d’items dans l’exercice de leurs fonctions. Ces items incluent par exemple des pansements, des tubes, des seringues, des gants, etc. Tous ces items doivent être disponibles en tout temps, sans quoi la qualité des soins de santé fournis aux patients est affectée.

Plusieurs systèmes de gestion d’inventaire, déjà en place aujourd’hui, tentent de faire le suivi de ces différents items. Cependant, ces solutions comportent certains problèmes, surtout si elles ne sont pas utilisées adéquatement par le personnel. Par exemple, une solution élaborée par l’entreprise Logi-D fait l’usage d’un système d’étiquettes RFID. Ces étiquettes doivent être déplacées entre les cabinets d’entreposage et les tableaux de commande afin de déclencher des demandes de réapprovisionnement.

Bien qu’efficace et en place depuis un certain temps, cette solution de Logi-D dépend de procédures strictes de la part du personnel de soins de santé. Le personnel doit manipuler les étiquettes RFID de façon ordonnée. Ce système requiert de prendre la bonne étiquette correspondant au bon item, ce qui peut être difficile dans une situation urgente. Ceci est une source d’inefficacité et parfois une cause d’erreur.

Logi-D propose d’améliorer ce système en remplaçant les étiquettes RFID par un système de surveillance d’inventaire avec caméra. Un tel système de vision artificielle permettrait de repérer les items manquants sans avoir à dépendre d’une action de la part du personnel de soins de santé. À l’aide d’une caméra placée au plafond, le système tente de reconnaître les compartiments de différents tiroirs d’entreposage et d’en évaluer leurs contenus. Un tel système pourrait améliorer significativement la manière avec laquelle les hôpitaux et cliniques médicales gèrent l’utilisation quotidienne des fournitures médicales. Ceci permettrait d’offrir une meilleure gestion d’inventaire dans les hôpitaux et un service de santé de meilleure qualité.

Une revue de littérature nous permet d’abord d’identifier des projets similaires à celui-ci. Ensuite, nous présentons notre solution au problème de détection d’inventaire. Ce problème peut être divisé en trois sous-problèmes à résoudre indépendamment : la détection et l’identification du tiroir, la détection et la localisation des compartiments, puis la détection des contenus des compartiments. Chacun de ces trois sous-problèmes est résolu séparément, selon différents algorithmes de traitement d’images.

Un premier système a pu être installé à l’Hôpital du Sacré-Cœur de Montréal en tant que projet pilote. Ceci nous a permis d’évaluer la performance et la fiabilité du système dans des conditions d’utilisation réelle durant quelques jours. Ainsi, parmi les 59 détections qui

auraient dû être faites, 45 ont été détectées par notre système, c'est-à-dire un taux de réussite de 76,3%. Les causes d'erreurs du système ont aussi pu être catégorisées et classées selon leurs occurrences. Ceux-ci incluent les problèmes d'occlusion, les tiroirs qui ne sont pas ouverts suffisamment et les changements d'éclairage.

Ce projet nous a permis de développer un système fonctionnel de détection d'inventaire. Par contre, certaines conditions d'utilisation doivent être remplies afin d'optimiser la performance du système. Pour pallier ces limitations, certaines améliorations pourraient être faites dans le futur. Un système de rétroaction pourrait permettre à l'utilisateur de savoir si la détection a été faite ou non. Aussi, l'utilisation des informations provenant d'une image de profondeur pourrait permettre de mieux détecter le contenu des compartiments.

ABSTRACT

Inventory management in hospitals can be a major logistical issue. Hospital staff depends on daily use of hundreds of items in the exercise of their functions. These include items such as bandages, tubes, syringes, gloves, etc. All these items should be available at all times, otherwise the quality of health care provided to patients is affected.

Several inventory management systems already in place today, are trying to keep track of these items. However, these solutions have some problems, especially if they are not used appropriately by staff. For example, a solution developed by the company Logi-D is using an RFID tag system. These tags must be moved between storage cabinets and control panels to trigger replenishment requests.

Although effective and in place for some time, this solution by Logi-D depends on strict procedures on the part of health care staff. Staff must handle RFID tags in an orderly fashion. This system requires taking the right tag corresponding to the correct item, which can be difficult in an emergency situation. This is a source of inefficiency and sometimes a cause of error.

Logi-D proposes to improve this system by replacing RFID tags with an inventory monitoring system, using a camera. Such an artificial vision system would identify the missing items without having to depend on actions by staff health care. Using a camera mounted on the ceiling, the system attempts to recognize the different storage compartments of drawers and tries to evaluate their contents. Such a system could significantly improve the way in which hospitals and medical clinics manage the daily use of medical supplies. This would provide better inventory management in hospitals and health service quality.

A literature review first allows us to identify projects similar to this one. Then we present our solution to inventory detection. This problem can be divided into three sub-problems to be solved independently: the detection and identification of the drawer, the detection and localization of the drawer compartments, and the detection of contents of the compartments. Each of these three sub-problems are solved separately by different image processing algorithms.

A first system has been installed at the Hôpital du Sacré-Cœur de Montréal as a pilot experiment. This allowed us to assess the performance and reliability of the system under actual conditions of use for several days. Thus, among the 59 possible detections that could have been made, 45 were detected by our system, that is to say, a success rate of 76.3%. Causes of system errors have also been categorized and classified according to their occurrences. These include occlusion problems, drawers which are not sufficiently open, and lighting changes.

This project has enabled us to develop a functional system of inventory detection. However, some conditions must be met in order to optimize system performance. To overcome these limitations, some improvements could be made in the future. A feedback system could provide the user with an inventory detection status. Also, the use of information from a depth image could help better detect the contents of the compartments.

TABLE DES MATIÈRES

REMERCIEMENTS	iii
RÉSUMÉ	iv
ABSTRACT	vi
TABLE DES MATIÈRES	viii
LISTE DES TABLEAUX	xi
LISTE DES FIGURES	xii
CHAPITRE 1 INTRODUCTION	1
1.1 Mise en contexte	1
1.2 Motivation	2
1.3 Objectifs de recherche	6
1.4 Organisation du mémoire	6
CHAPITRE 2 REVUE DE LITTÉRATURE	8
2.1 Projets d’inspection d’inventaires de commerce au détail	8
2.2 Brevets d’inspection d’inventaires de commerce au détail	10
2.3 Autres projets semblables dans un contexte domestique	11
2.4 Projets incluant la détection de tiroirs	13
2.5 Résumé de la revue de littérature	15
CHAPITRE 3 PREMIER THÈME - DÉTECTION DE TIROIR	17
3.1 Correction de la distorsion	17
3.1.1 Choix de la caméra	17
3.1.2 Distorsion radiale	17
3.1.3 Distorsion tangentielle	20
3.1.4 Estimation des paramètres de distorsion	20
3.2 Détection de marqueurs de tiroir	21
3.2.1 Forme des marqueurs	21
3.2.2 Segmentation de marqueurs	22
3.2.3 Identification des paires de marqueurs	23

3.2.4	Limitations de la détection de marqueurs	24
3.3	Identification de tiroir par les marqueurs	25
3.3.1	Choix de la méthode d'identification	25
3.3.2	Limitations	26
3.4	Détection de tiroir par les côtés	26
3.4.1	Caractéristiques des tiroirs	27
3.4.2	Soustraction d'arrière-plan	28
3.4.3	Détection de lignes	29
3.5	Identification de tiroir par les côtés	34
3.6	Conclusion	35
CHAPITRE 4	SECOND THÈME - DÉTECTION DE COMPARTIMENTS	36
4.1	Détection de séparateurs par transformée de Hough	37
4.1.1	Justification	37
4.1.2	Détection de contours	38
4.1.3	Transformée de Hough en niveaux de gris	39
4.1.4	Choix de la taille de l'espace de Hough	40
4.1.5	Identification des maxima de l'espace de Hough	41
4.1.6	Détection de lignes horizontales	44
4.1.7	Délimitation des compartiments à partir des séparateurs	44
4.2	Utilisation de la calibration pour détecter les séparateurs	45
4.2.1	Vue d'ensemble	45
4.2.2	Détails d'implémentation	46
4.2.3	Procédure de recalage	46
CHAPITRE 5	TROISIÈME THÈME - CLASSIFICATION DES CASIERS	48
5.1	Conditions préalables à la classification	49
5.2	Classification des casiers selon les contours d'items	50
5.2.1	Justification	50
5.2.2	Explication de l'algorithme	50
5.2.3	Limitations	51
5.3	Détection du contenu par la technique de croissance de région	52
5.3.1	Justification	52
5.3.2	Explication de l'algorithme	52
5.3.3	Limitations	53
5.4	Classification des casiers par soustraction d'arrière-plan	54
5.4.1	Conditions préalables	54

5.4.2	Exemple de soustraction d'arrière-plan	55
5.4.3	Recalage des images	56
5.4.4	Limitations	60
5.5	Classification des casiers par détection de fonds quadrillés	61
5.5.1	Justification	61
5.5.2	Détection de carrés par transformée de Hough généralisée	62
5.5.3	Autres méthodes de détection de carrés	62
5.5.4	Extraction de contours	64
5.5.5	Détection des segments de carrés par l'algorithme de Burns	65
5.5.6	Combinaisons des segments détectés en carrés	68
5.5.7	Croissance de région à partir des carrés détectés	68
5.5.8	Limitations	69
5.6	Détection du contenu par segmentation de couleur	70
5.6.1	Justification	70
5.6.2	Choix de la couleur du fond	71
5.6.3	Choix d'un espace de couleur	71
5.6.4	Segmentation par seuillage	72
5.6.5	Résultat de la segmentation	73
5.6.6	Classification à partir de la segmentation	74
5.6.7	Avantages et inconvénients	75
CHAPITRE 6	RÉSULTATS ET DISCUSSION	76
6.1	Introduction	76
6.2	Méthodes de détection de tiroir	77
6.2.1	Mise en contexte	77
6.2.2	Résultats et discussion	79
6.3	Comparaison avec le système "2Bin-iD" de l'Hôpital du Sacré-Cœur	81
6.3.1	Mise en contexte	81
6.3.2	Résultats et discussion	83
6.3.3	Comparaison de l'efficacité des deux systèmes	88
CHAPITRE 7	CONCLUSION	89
7.1	Synthèse des travaux	89
7.2	Limitations de la solution proposée	90
7.3	Améliorations futures	91
RÉFÉRENCES	94

LISTE DES TABLEAUX

Tableau 2.1	Tableau résumant la revue de littérature	16
Tableau 6.1	Exemple de compilation des détections de tiroirs pour une période de 16 jours	79
Tableau 6.2	Nombre d'évènements correspondants pour la période du 4 février au 5 mars 2013	85
Tableau 6.3	Causes des conflits entre les deux systèmes	86
Tableau 6.4	Calcul du nombre réel d'évènements filmés	88

LISTE DES FIGURES

Figure 1.1	Circulation des produits à travers l'hôpital	2
Figure 1.2	Un cabinet typique du système "2Bin-iD".	3
Figure 1.3	Un tiroir typique du système "2Bin-iD" comprenant 2 séparateurs et 4 casiers.	4
Figure 1.4	Cycle de réapprovisionnement du système "2Bin-iD" avec un tableau RFID.	5
Figure 3.1	Image de la caméra avec distorsion en barillet	18
Figure 3.2	Image de la caméra avec correction de la distorsion radiale avec K_1 et K_2	19
Figure 3.3	Image de la caméra avec correction de la distorsion radiale avec K_1 , K_2 et K_3	19
Figure 3.4	Positions et segmentations des marqueurs verts fluorescents.	21
Figure 3.5	Composantes Cb et Cr de l'espace YCbCr.	23
Figure 3.6	Effet de perspective de la caméra	26
Figure 3.7	Un exemple de tiroir incliné.	27
Figure 3.8	Exemple de soustraction d'arrière-plan	29
Figure 3.9	Exemple d'extraction des contours.	30
Figure 3.10	Les paramètres ρ et θ d'une droite	31
Figure 3.11	Deux points A et B dans une image	32
Figure 3.12	L'espace de Hough	33
Figure 3.13	Accumulateur pour l'algorithme de Hough	34
Figure 3.14	Transformée de Hough	35
Figure 3.15	Intersections entre les côtés du tiroir et les côtés du cabinet	35
Figure 4.1	Image d'un tiroir avec 5 séparateurs : 3 séparateurs horizontaux et 2 verticaux	36
Figure 4.2	Image du tiroir avec détection de contours verticaux par une convolution de Sobel	39
Figure 4.3	Espace de Hough de complet	41
Figure 4.4	Espace de Hough réduit aux lignes verticales avec identification des maxima	42
Figure 4.5	Lignes détectées dans l'image du tiroir par l'algorithme de Hough	43
Figure 4.6	Exemple de délimitation des compartiments à partir des séparateurs	45
Figure 4.7	Illustration de quelques mesures faites durant la calibration	46

Figure 4.8	Exemple de délimitation des casiers à partir des marqueurs	47
Figure 5.1	Images des différents types de fond de tiroir possibles	48
Figure 5.2	Tiroir avec 2 casiers vides et 10 casiers pleins	51
Figure 5.3	Exemple de casier contenant des boîtes d'items	52
Figure 5.4	Progression de l'algorithme de croissance de région sur l'image d'un casier après différentes itérations. La région est illustrée en vert. . . .	53
Figure 5.5	Images utilisées pour effectuer la soustraction d'arrière-plan d'un compartiment avec fond lisse	55
Figure 5.6	Exemple de transformation en perspective d'une image de compartiment	56
Figure 5.7	Exemple de transformation en perspective d'une image quadrillée. Les 4 coins numérotés servent à trouver la matrice de transformation. . .	58
Figure 5.8	Images utilisées pour effectuer la soustraction d'arrière-plan d'un compartiment avec fond quadrillé	60
Figure 5.9	Exemple de casier avec un fond de type quadrillé	61
Figure 5.10	Étapes de détection des lignes verticales et horizontales	65
Figure 5.11	Partitionnement des gradients selon leur orientation par l'algorithme de Burns	66
Figure 5.12	Détection des segments de lignes par Burns et identification des carrés par des points verts	67
Figure 5.13	Exemple de casier avec un fond en partie quadrillé	69
Figure 5.14	Détection de carrés (en rouge) suivie d'une croissance de région (en rose)	70
Figure 5.15	Exemple de conversion à l'espace de couleur RGB normalisé	73
Figure 5.16	Exemple de segmentation d'un fond orange	74
Figure 6.1	Exemples de bandes utilisées pour la recherche de marqueurs	78
Figure 6.2	Exemple de graphique créé à partir des mesures d'un seul compartiment sur plusieurs trames vidéo	84
Figure 6.3	Occurrences de chaque type d'erreur du système par caméra	87

CHAPITRE 1

INTRODUCTION

1.1 Mise en contexte

Les soins de santé administrés par les hôpitaux nécessitent que plusieurs produits médicaux soient disponibles en tout temps, sans quoi la qualité des soins en serait affectée. Le personnel infirmier doit utiliser de nombreux produits tels que des pansements, des gants, des seringues, des tubes, des solutés et plusieurs autres. Le grand nombre de produits différents requis fait en sorte qu'une bonne gestion de cet inventaire devient une condition nécessaire à la qualité des soins. Plusieurs problèmes de logistiques peuvent alors se poser afin d'améliorer l'efficacité de la chaîne de réapprovisionnement de ces produits.

Cette chaîne de réapprovisionnement d'un hôpital peut normalement être divisée en deux différents niveaux ou paliers (figure 1.1). En haut de la chaîne se trouve le magasin général qui est en fait un grand entrepôt par où tous les produits arrivent à l'hôpital. Un magasinier s'occupe ensuite de prendre certains produits du magasin général pour les distribuer à différents services ou départements à l'intérieur de l'hôpital. Ces services peuvent être par exemple, le service d'orthopédie, de chirurgie générale ou de cardiologie. Chacun de ces services s'occupe d'entreposer ses produits à sa façon dans une petite réserve. C'est enfin à partir de cette réserve que le personnel infirmier peut utiliser différents produits nécessaires aux soins pour les patients.

Cette chaîne de réapprovisionnement permet normalement une bonne circulation des produits à travers l'hôpital. On suppose que le magasinier fait une bonne distribution des produits et que les commandes de réapprovisionnement sont bien envoyées au magasin général lorsqu'il le faut. En effet, lorsqu'un certain type de produit devient manquant dans une réserve, c'est la responsabilité du département de commander de nouveaux produits au magasin général. Le problème est qu'un produit peut facilement tomber en rupture de stock dans une réserve si l'on oublie d'en faire la commande au magasin. Afin d'éviter ce genre de problème, plusieurs systèmes de gestion d'inventaire, déjà en place aujourd'hui, tentent de faire le suivi des différents produits utilisés par le personnel infirmier à l'intérieur des réserves.

L'entreprise Logi-D de Laval se spécialise justement dans la création de tels systèmes de gestion d'inventaire. L'entreprise offre des solutions d'automatisation de la chaîne logistique pour les hôpitaux. Ses solutions visent évidemment à être efficaces, à faciliter la gestion

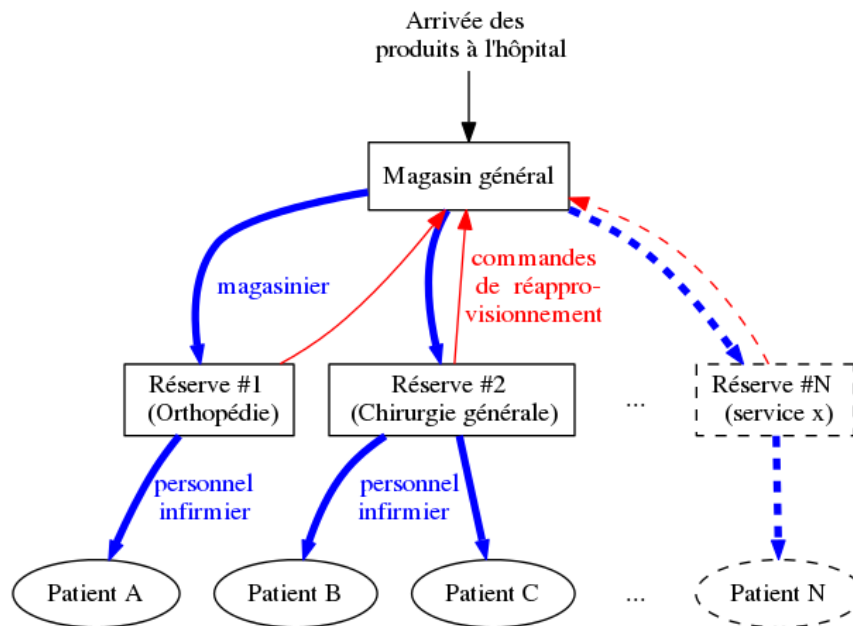


Figure 1.1 Circulation des produits à travers l'hôpital

d'inventaire et à améliorer la chaîne de réapprovisionnement de produits que nous avons présentée ici.

1.2 Motivation

Comme la gestion d'un grand inventaire de produits peut devenir compliquée, il peut être avantageux pour un hôpital d'utiliser un système de gestion informatisé. Le choix d'un système de gestion approprié est important pour un hôpital puisqu'il peut avoir un impact significatif sur la disponibilité ou non-disponibilité des ressources de soins de santé. Un tel système peut s'occuper de gérer les produits de façon automatique ou semi-automatique avec l'aide du personnel. Il existe plusieurs systèmes de gestion, déjà en place à ce jour, qui tentent de faire un suivi des différents produits utilisés. Cependant, ces solutions comportent souvent certaines limites, surtout si elles ne sont pas utilisées adéquatement par le personnel.

Dans notre cas, on s'intéresse en particulier au système de gestion nommé "2Bin-iD" élaboré par l'entreprise Logi-D. Cette entreprise offre d'installer des cabinets pouvant contenir environ une dizaine de tiroirs chacun tel que montré dans la figure 1.2. Chaque tiroir est divisé en plusieurs "compartiments" ou "casiers" par des séparateurs ayant des positions ajustables comme dans la figure 1.3. Le but de cet ensemble de cabinets et de tiroirs est d'offrir des espaces de rangement à tous les produits pouvant être utiles au personnel infirmier.



Figure 1.2 Un cabinet typique du système “2Bin-iD”.
(Image reproduite avec l’autorisation de Logi-D.)

Au-delà de l’aspect physique des cabinets, l’innovation du système “2Bin-iD” est d’utiliser des étiquettes RFID afin d’identifier chaque type de produit à commander. Le système comprend une série d’étiquettes RFID et une correspondance “un-à-un” est établie entre chaque étiquette et chaque produit de l’inventaire. L’étiquette d’un produit est normalement accrochée au casier du produit lorsque sa quantité est suffisante. Lorsque le produit doit être commandé, il suffit de décrocher son étiquette du casier, et de la poser sur un petit tableau lecteur de RFID qui se trouve posé contre un mur. Ceci a pour effet d’initier une nouvelle commande qui est envoyée au magasin général par le réseau informatique local de l’hôpital. Les différentes étapes de ce cycle de réapprovisionnement sont illustrées dans la figure 1.4.

On appelle ce système “2Bin-iD” parce que chaque type de produit est en fait entreposé dans deux casiers : un casier primaire et un casier secondaire. Pour utiliser le système, le personnel doit effectuer quelques opérations de façon ordonnée. D’abord, le personnel infirmier doit consommer les produits uniquement du casier primaire, jusqu’à ce que ce casier soit complètement vide, avant de pouvoir consommer ceux du casier secondaire. Lorsqu’un des deux casiers se vide, il faut prendre l’étiquette RFID qui est attachée au panier pour la déposer sur un tableau de commande. Ce tableau peut alors faire la lecture de l’étiquette RFID pour initier la commande de réapprovisionnement. En attendant que la commande de réapprovisionnement soit traitée par le magasin, le personnel infirmier peut continuer de consommer des produits depuis le casier secondaire. Éventuellement, un magasinier décide d’effectuer l’activité de réapprovisionnement et apporte tous les produits demandés à la réserve.



Figure 1.3 Un tiroir typique du système “2Bin-iD” comprenant 2 séparateurs et 4 casiers.
(Image reproduite avec l’autorisation de Logi-D.)

Pour bien faire le réapprovisionnement, le magasinier doit effectuer une rotation de stock, c’est-à-dire qu’il doit transférer tous les produits restants de la case secondaire vers la case primaire. Ensuite, il place les nouveaux produits dans la case secondaire. Cette rotation permet d’éviter que certains produits de la case secondaire ne deviennent périmés. Finalement, le magasinier termine son activité de réapprovisionnement en prenant les étiquettes RFID qui ont été placées sur le tableau de commande et en les replaçant sur les casiers correspondants.

Le système “2Bin-iD” de Logi-D facilite beaucoup la gestion d’inventaire et c’est pourquoi de nombreux hôpitaux l’ont déjà adopté. Le fait d’avoir un double casier pour chaque produit encourage le personnel à demander un réapprovisionnement dès que le stock est à moitié écoulé. Avec un casier simple, le personnel pourrait attendre que le casier se vide avant d’envoyer des commandes de nouveaux produits en retard. Un autre avantage intéressant du système est l’usage des étiquettes RFID qui simplifient beaucoup l’envoi des commandes au magasin général. Pour déclencher des commandes, le personnel doit seulement manipuler des étiquettes, plutôt que d’avoir à saisir des commandes dans un ordinateur. Landry et Beaulieu (2010) donnent davantage de détails sur ce système et expliquent les différents bénéfices qu’il apporte.

Par contre, ce système présente certains inconvénients et certaines limitations. D’abord, comme nous l’avons mentionné, ce système requiert que des procédures strictes soient suivies par le personnel de soins de santé. Le personnel doit donc avoir été formé au préalable sur l’usage adéquat des étiquettes RFID. Même lorsque le personnel est bien formé, des erreurs de manipulation pourront toujours survenir. Par exemple, une étiquette peut être remplacée dans un mauvais tiroir ou bien se décrocher d’un panier et tomber au sol. Les erreurs sont encore plus probables lorsque le nombre d’étiquettes augmente. Normalement, on peut retrouver quelques centaines d’étiquettes par réserve, si l’on compte qu’il peut y avoir

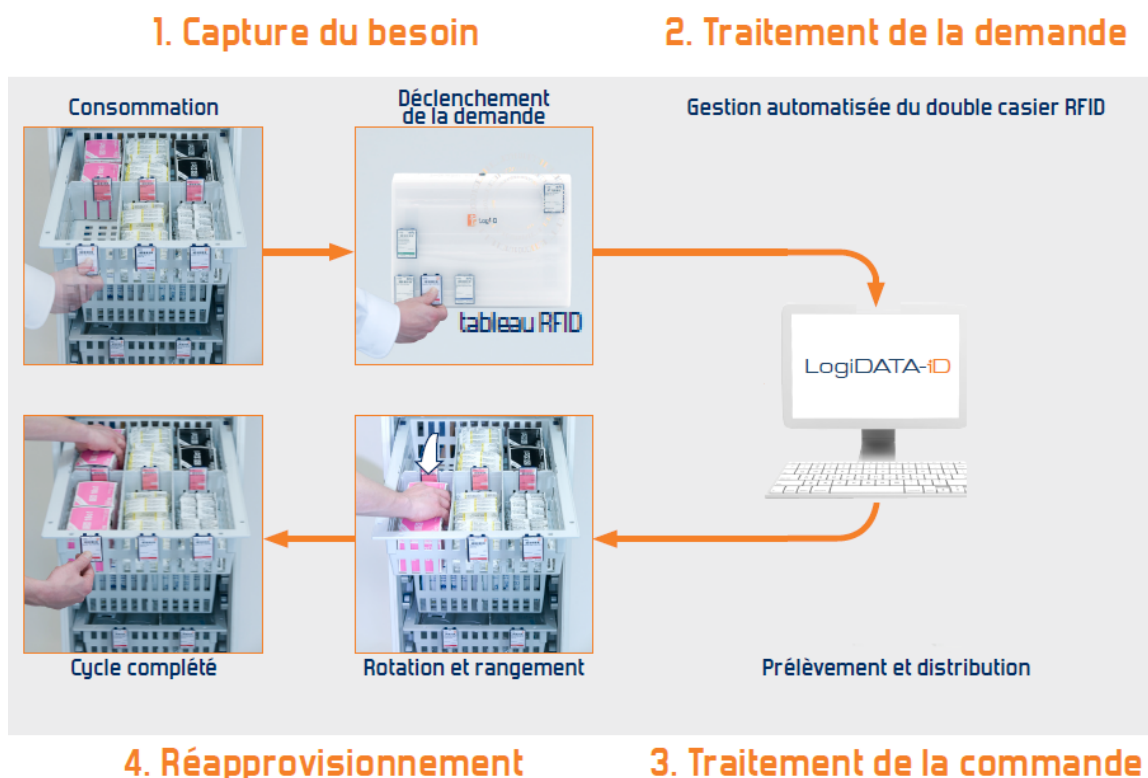


Figure 1.4 Cycle de réapprovisionnement du système "2Bin-iD" avec un tableau RFID.
(Image reproduite avec l'autorisation de Logi-D.)

environ 5 cabinets de 10 tiroirs par réserve, que chaque tiroir contient en moyenne 6 casiers de produits, et qu'il faut une étiquette par casier. Enfin, certains départements comme ceux offrant des soins d'urgence n'ont pas toujours le temps d'effectuer toutes les manipulations d'étiquettes nécessaires au système.

Il serait avantageux d'éviter tous ces inconvénients en réduisant le nombre de manipulations nécessaires au système. Une solution envisagée par Logi-D, est de remplacer le système d'étiquettes RFID par une caméra qui surveille l'inventaire. Avec un système de vision artificielle de surveillance d'inventaire, il serait possible de repérer les items manquants, et de déclencher automatiquement les commandes de réapprovisionnement. Bref, ce système permettrait d'éliminer les manipulations d'étiquettes RFID et tous les inconvénients que nous avons mentionnés précédemment. Les hôpitaux sont aussi intéressés par un tel système puisqu'il permettrait de faciliter la tâche de tout le personnel infirmier et d'offrir un service de santé de meilleure qualité.

Dans cette recherche, on essaie donc d'améliorer ce système de gestion d'inventaire en le remplaçant par un système de surveillance d'inventaire avec caméra. Si un tel système peut fonctionner, cela pourrait améliorer significativement la manière avec laquelle les hôpitaux

et cliniques médicales gèrent l'utilisation de leurs fournitures médicales. On éviterait aussi toutes les erreurs dues aux manipulations d'étiquettes. Enfin, un tel système d'inventaire pourrait aussi bien être appliqué dans un autre contexte tel qu'une manufacture, un entrepôt ou encore un commerce au détail.

1.3 Objectifs de recherche

L'objectif général de cette recherche est de sélectionner, d'analyser, d'évaluer et de comparer différents algorithmes de vision par ordinateur, afin de faire le suivi automatique de l'inventaire médical et d'en repérer les items manquants. Le problème principal est de traiter correctement les images envoyées par la caméra afin de pouvoir établir correctement l'état de l'inventaire. Ce système permettrait de remplacer le système actuel de puces RFID. Pour résoudre ce problème, on identifie les objectifs de recherche suivants :

1. Développer des algorithmes de traitement d'images pour faire la détection et l'identification d'un tiroir ouvert.
2. Développer des algorithmes de traitement d'images pour localiser les différents compartiments d'un tiroir donné.
3. Développer des algorithmes de traitement d'images pour faire le suivi des changements d'état des compartiments, les états possibles étant vide ou non-vide.
4. Implémenter la méthode de traitement choisie et en faire l'installation dans un hôpital.
5. Vérifier et valider la méthode proposée à l'aide d'une banque de vidéos provenant d'un hôpital. Évaluer la performance et robustesse des algorithmes.

1.4 Organisation du mémoire

Ce mémoire est organisé selon le modèle classique : on y retrouve les chapitres habituels d'introduction, de revue de littérature, de méthodologie, d'analyse et de discussion des résultats puis de conclusion. D'abord, la revue de littérature nous permet de vérifier s'il n'existe pas de projets de recherche semblables et d'identifier des algorithmes de traitement d'images qui pourraient être appropriés au problème de détection visuelle d'inventaire.

Ensuite, dans la section méthodologie, nous décrivons la méthode choisie pour faire la détection de l'inventaire. On peut diviser notre méthodologie en trois sous-problèmes à résoudre indépendamment. En premier lieu, le système doit pouvoir reconnaître un tiroir lorsqu'il est ouvert. Ensuite, une fois le tiroir repéré, il faut que chacun des compartiments du tiroir soit délimité. Finalement, le système doit pouvoir évaluer la quantité de produits de chacun des compartiments pour établir l'état de l'inventaire.

Des algorithmes seront testés pour chacun de ces trois sous-problèmes. Une méthode formelle d'évaluation sera appliquée sur ces algorithmes afin d'identifier les algorithmes les plus performants. Le critère d'évaluation sera le taux de réussite sur une banque d'images et de vidéos.

Une fois la méthodologie décrite, nous présenterons et analyserons la performance de notre méthode dans la section de discussion des résultats. Finalement, dans le dernier chapitre, nous ferons la conclusion de ce travail et nous présenterons quelques perspectives de recherche possibles pour le futur.

CHAPITRE 2

REVUE DE LITTÉRATURE

Dans cette revue de littérature, nous faisons un inventaire de travaux antérieurs portant sur des projets reliés à l’inspection automatisée d’inventaires à l’aide de vision artificielle. En effet, la littérature scientifique nous permet de trouver des projets de recherche similaires au nôtre. Cependant, aucun de ces projets n’était vraiment semblable au nôtre. Il peut néanmoins être utile de s’intéresser à ces projets, puisque certaines de leurs idées pourraient nous aider.

2.1 Projets d’inspection d’inventaires de commerce au détail

D’abord, deux projets semblables au nôtre sont des projets de conception de robot visant à faire l’inspection de l’inventaire de commerces de détail. Le premier projet, nommé “AndyVision” (Mankodiya *et al.*, 2012) concerne plus particulièrement l’inventaire des commerces de détail. Dans ce type de commerce, le personnel doit normalement s’assurer que les produits de chaque étagère sont bien placés et qu’aucun produit n’est épuisé. Si ce genre de tâche pouvait être effectué par un robot, ceci permettrait aux magasins de réaliser des économies appréciables.

Pour ce faire, Mankodiya *et al.* (2012) proposent de concevoir un robot capable de naviguer à l’intérieur d’un magasin de façon autonome. La navigation du robot est facilitée par le positionnement de différentes antennes Wi-Fi à des endroits stratégiques à l’intérieur du magasin. On peut alors déduire la position du robot en analysant la force des signaux de ces antennes. Lorsque le robot arrive devant l’étagère voulue, ses caméras lui permettent de photographier l’étagère et d’envoyer ces images à un serveur. Ce serveur peut alors traiter les images d’étagères et utiliser des techniques de reconnaissance d’objet comme la comparaison à des modèles standards (template matching), ou encore la reconnaissance optique de caractères. Il est plus facile d’effectuer ces techniques du côté du serveur, puisqu’on peut alors avoir un meilleur accès à une banque de produits et d’images de références.

Une fois les objets reconnus, il serait alors possible de générer, en temps réel, un planogramme correspondant à la disposition actuelle des produits. Un planogramme permet de visualiser le placement de chaque produit et de trouver leur disposition optimale. Ceci permettrait aussi d’identifier les produits manquants ou mal placés.

Le projet “AndyVision” comporte certaines ressemblances avec le nôtre puisqu’il s’intéresse lui aussi à la surveillance d’inventaire. Cependant, le type d’inventaire surveillé n’est pas le même puisque “AndyVision” se concentre sur le commerce de détail alors que nous nous intéressons à l’inventaire médical. Dans le commerce de détail, on ne retrouve normalement pas de tiroir pour ranger les produits, mais plutôt des étagères. De plus, les emballages de produits commerciaux diffèrent des emballages médicaux.

Il aurait été intéressant d’avoir des résultats de la performance “AndyVision”, mais l’article nous indique que le projet était encore à la phase de projet pilote. Selon les auteurs, il n’existe pour l’instant aucun robot qui permet de générer les planogrammes d’un magasin en temps réel.

Le deuxième projet, plus récent, s’intéresse lui aussi à la conception d’un robot pour la gestion d’inventaire dans le commerce de détail (Francis *et al.*, 2013). Ce robot, nommé “MetaBot”, pourrait accomplir différentes tâches programmées par l’utilisateur. Des exemples de tâches sont de naviguer jusqu’à un point donné du magasin, d’éviter un obstacle, de longer un mur, de détecter les produits manquants ou de vérifier l’espacement et l’alignement des produits d’une étagère. Par contre, ce projet ne parle pas de générer des planogrammes comme dans le cas de “AndyVision”.

Pour naviguer, “MetaBot” utilise plutôt une carte du magasin décrivant les positions des différentes étagères. Des capteurs de distance sont utilisés par le robot pour se repérer durant sa navigation. Dans le cas de “MetaBot”, la détection des ruptures de stock se fait plutôt par l’utilisation d’images de profondeur par une caméra comme la Kinect. Francis *et al.* (2013) donnent les détails de cet algorithme. Une fois l’image de profondeur obtenue, l’algorithme des K-moyennes est utilisé afin de partitionner les points en 3 groupes de distances correspondants à l’avant-plan, l’arrière-plan et une distance moyenne entre ces deux plans. Le groupe de points correspondant à l’avant-plan est ensuite segmenté de façon à ce que chaque segment représente une étagère et ses produits. Chaque étagère peut ensuite être traitée indépendamment des autres afin de déterminer si des produits sont manquants. Francis *et al.* (2013) supposent qu’un vide sera créé dans l’avant-plan de l’étagère lorsqu’un produit est absent. Si la taille de ce vide est plus grande qu’une certaine valeur trouvée empiriquement, le produit de cette étagère est alors considéré comme étant en rupture de stock.

Des tests des différents algorithmes de “MetaBot” ont été faits en laboratoire. En plus de tester la navigation du robot vers les étagères, les tests incluaient aussi l’aspect de détection d’inventaire. Des boîtes de produits, comme des céréales, étaient placées sur les étagères, puis le robot devait détecter correctement les produits manquants. Les résultats obtenus sont que les 6 produits manquants étaient toujours détectés correctement, mais que 2 fausses détections étaient aussi générées. Francis *et al.* (2013) affirment que ces fausses détections

sont dues au fait que les étagères utilisées dans les tests n’avaient pas de plaque de fond. Selon eux, ce problème ne devrait pas se produire dans des commerces réels puisque les étagères ont normalement des plaques de fond. Les auteurs rapportent être prêts à passer à des tests de performance dans des conditions réelles, mais les résultats de ces tests n’ont pu être trouvés.

2.2 Brevets d’inspection d’inventaires de commerce au détail

En plus de s’intéresser à la littérature scientifique, on peut aussi s’intéresser aux brevets déjà existants. Même si les brevets ne décrivent pas de démarches scientifiques complètes et qu’ils ne présentent aucun résultat, ils peuvent parfois nous permettre de comprendre comment d’autres systèmes semblables ont été pensés. Ici, nous nous intéressons à deux brevets en particulier puisqu’ils concernent tous les deux l’inspection d’inventaire de commerce au détail.

Le premier brevet (Schwartz, 2007) décrit un système de caméras qui se trouvent dans un commerce au détail et qui visent à faire le suivi des produits lorsqu’ils sont manipulés par les clients. Lorsqu’un produit est retiré d’une étagère et qu’il est placé dans un panier d’achat par un client, des caméras positionnées dans le magasin permettraient alors d’identifier à la fois le client et les produits qu’il a sélectionnés. De cette façon, le système peut conserver en mémoire la liste des produits qui se trouvent dans les paniers de chaque client. Le système doit aussi reconnaître un produit lorsqu’il est retiré d’un panier, puis replacé sur une étagère.

Grâce à ce suivi de produits, il est possible de demander à tout moment la liste des produits qui se trouvent dans le panier d’un client. Un dispositif d’affichage placé sur le panier d’achat pourrait ainsi permettre au client de consulter la liste de ses produits, ainsi que le montant total. Éventuellement, lorsqu’un client termine de magasiner et qu’il arrive à un terminal de point de vente, il ne serait pas nécessaire de vider le contenu du panier, puisque le système saurait déjà la liste des produits du client. Il suffirait pour le client d’effectuer son paiement en laissant tous ses produits à l’intérieur du panier.

Le système décrit par ce brevet serait avantageux puisqu’il permettrait d’éviter les files d’attente aux points de vente, en plus de réduire les coûts associés à l’embauche du personnel. Par contre, la présence de ce brevet ne signifie pas qu’un tel système a réellement été implémenté dans un magasin. En effet, un tel système serait sûrement très difficile à implémenter à cause des problèmes d’occlusions qui pourraient survenir pendant la manipulation des produits par les clients. Même si plusieurs caméras filmaient en même temps un même client malveillant lorsqu’il prend un produit d’une étagère, il serait toujours possible pour ce client de cacher des produits de petite taille pour qu’ils ne soient pas reconnus par le système. Heureusement pour notre projet, dans un contexte d’inventaire médical, on peut

supposer que le personnel infirmier n’aura aucun intérêt à cacher les produits utilisés, et que les utilisateurs du système seront généralement coopératifs.

Le deuxième brevet (Opalach *et al.*, 2012) décrit un système un peu semblable au premier brevet. Ce système se situe aussi dans un commerce au détail afin de surveiller le niveau d’inventaire des produits. Par contre, le but du système n’est pas d’obtenir une liste des produits choisis par les clients, mais plutôt de déterminer la quantité de produits encore présents sur les étagères. Les caméras sont donc placées de façon à filmer uniquement les étagères de face.

Une fois que les images ou les vidéos sont enregistrées par une caméra, elles sont envoyées à un serveur qui s’occupe d’en faire le traitement par des techniques d’analyse d’image. Les produits présents dans ces images peuvent alors être identifiés par une technique de reconnaissance d’objet, par exemple, en utilisant des points caractéristiques comme ceux obtenus par la technique de SIFT (“scale-invariant feature transform”) créé par Lowe (2004). Le brevet n’exclut pas non plus la possibilité d’utiliser d’autres algorithmes d’analyse d’image afin de déterminer le statut de l’inventaire. Une possibilité, par exemple, serait de poser des motifs au fond des étagères. Lorsque les produits seraient retirés de l’étagère, ces motifs deviendraient alors visibles par le système qui pourrait ainsi estimer la quantité restante de produits. Enfin, l’identification des différents produits pourrait également permettre la génération de planogrammes pour ces étagères. Encore une fois, ce brevet ne donne malheureusement aucun résultat, mais nous permet toutefois d’identifier quelques idées d’algorithmes qui pourraient s’appliquer à notre projet aussi.

2.3 Autres projets semblables dans un contexte domestique

Les projets vus jusqu’à présent s’appliquaient tous à des contextes de commerce au détail. En plus des commerces au détail, on peut aussi considérer les projets s’appliquant à un contexte domestique. Nous présenterons deux de ces projets dans cette section. Le premier projet concerne la détection du contenu d’un réfrigérateur, alors que le deuxième porte sur la détection du niveau de déchets ménagers au moment de leur collecte par un camion à ordures.

La multinationale suédoise “AB Electrolux”, qui se spécialise dans les électroménagers, s’est intéressée à la détection du contenu d’un réfrigérateur domestique. Un brevet a été déposé par cette compagnie Roseen (2000). Ce brevet décrit comment des caméras sont placées à l’intérieur d’un réfrigérateur afin d’identifier les produits présents. Un dispositif d’affichage permettrait également d’afficher à l’utilisateur les images prises par les caméras, ainsi que la liste des produits détectés, et la liste des produits manquants.

Une recherche correspondant à ce brevet a été faite. L'article de cette recherche (Färnström *et al.*, 2002) permet d'avoir un peu plus de détails sur les algorithmes utilisés. D'abord, une détection de mouvement est effectuée afin de déterminer les moments où les produits sont manipulés. La détection de mouvement se fait à partir de soustractions de trames consécutives. Si la différence est plus grande qu'un certain seuil fixé à l'avance, on considère alors qu'il y a eu du mouvement. En sélectionnant seulement les trames avant et après le mouvement, il est possible de déterminer le type d'action effectué. On définit les actions possibles comme étant l'ajout, le retrait ou le déplacement de produits.

Afin de pouvoir identifier les produits plus facilement, Färnström *et al.* (2002) propose de placer une caméra sous les étagères transparentes, en filmant vers le haut. Ceci permet de voir les empreintes laissées par les produits sur ces étagères afin d'effectuer une meilleure identification. Par contre, les produits ne doivent pas être empilés, mais placés directement sur l'étagère. L'article décrit ensuite comment il est possible de déduire le type d'action effectué en comparant les images d'empreintes avant et après qu'un mouvement ait été détecté. Selon la différence de deux images d'empreintes, on détermine ainsi s'il s'agit d'un ajout, d'un retrait ou d'un déplacement de produit.

L'article ne donne cependant aucun résultat et indique seulement que d'autres travaux devront être effectués. Notre projet pourrait s'inspirer des idées présentées. Par contre, comme nos tiroirs ne sont pas transparents, il serait difficile d'obtenir une image d'empreinte des produits. Ceci impliquerait de remplacer tous les tiroirs déjà présents dans les hôpitaux par des tiroirs transparents et nous préférierions trouver une solution moins dispendieuse.

Un autre projet de recherche concerne la détection du niveau de déchets ménagers au moment de leur collecte par un camion à ordures. Cette même recherche est présentée dans deux articles (Hannan *et al.*, 2011) et (Hannan *et al.*, 2012). Le but de cette recherche est de pouvoir identifier la quantité de déchets ménagers au moment de leur collecte. Cette information peut être utile pour différentes raisons. Entre autres, ceci permettrait une meilleure planification du trajet pendant la collecte des ordures. Pour ce faire, une caméra est d'abord placée à l'arrière du camion à ordures de façon à pouvoir filmer de haut les contenants de déchets.

Le problème principal de ce projet est de classifier les images d'ordures en différentes classes selon la quantité d'ordures présentes. L'auteur définit 5 niveaux possibles : bas, médium, rempli, excédant et débordement. Un descripteur de texture particulier est utilisé pour décrire les images d'ordures. Il s'agit de la matrice GLAM (pour "Gray Level Aura Matrix") définie pour la première fois par Qin et Yang (2005). Cette matrice indique, pour chaque niveau de gris, la probabilité qu'un autre niveau de gris soit présent dans les pixels avoisinants. Pour une image binaire, la matrice GLAM aura seulement une taille de 2

colonnes par 2 rangées, mais pour une image contenant 256 niveaux de gris, les dimensions de cette matrice seront de 256 par 256. Qin et Yang (2005) donnent plus de détails sur la façon de calculer cette matrice.

Une fois la matrice GLAM obtenue, les éléments de cette matrice sont donnés à un classifieur qui détermine un niveau de quantité d’ordures parmi les 5 niveaux possibles. Deux classifieurs sont testés : le premier est un classifieur neuronal de type perceptron multicouche et le deuxième est la méthode des K plus proches voisins. Ces deux classifieurs requièrent une phase d’apprentissage à partir d’une banque d’images. En choisissant de bons paramètres pour le calcul de la matrice GLAM, Hannan *et al.* (2012) parviennent à atteindre une efficacité de plus de 95% dans leur classification. Ces résultats sont très convaincants. Par contre, le seul inconvénient à cette méthode est qu’une phase d’apprentissage est nécessaire afin d’entraîner les classifieurs.

2.4 Projets incluant la détection de tiroirs

Dans cette section, nous nous intéressons au problème de détection de tiroirs. Comme un aspect important de notre projet sera la détection des tiroirs des cabinets, il est utile de faire une revue de la littérature sur ce sujet. Deux articles que nous avons trouvés concernent justement la détection de tiroirs. Le premier article présente une méthode de détection d’ouverture de tiroir de caisse enregistreuse. Le deuxième article concerne la détection et l’ouverture des tiroirs dans une cuisine par un robot.

Mustafa et Sethi (2005) présentent un système de reconnaissance d’objet en mouvement basé sur la détection des côtés de l’objet. Ce système peut s’appliquer à la détection de chariots dans un commerce, mais aussi à la détection du tiroir d’une caisse enregistreuse. Leur algorithme commence par effectuer une soustraction d’arrière-plan. Comme avec plusieurs techniques de soustraction, on suppose que la caméra reste fixe. Leur technique de soustraction effectue une mise à jour de l’arrière-plan afin de tenir compte des changements d’illumination de la scène. Une fois la soustraction faite, on obtient des pixels clairs là où il y a eu le plus de mouvement et des pixels plus sombres autrement.

Ensuite, le résultat de la soustraction donne une image en niveaux de gris sur laquelle est appliqué un filtre laplacien. Ce filtre permet de mettre en valeur les contours des objets. Par contre, l’image filtrée peut contenir un peu de bruit à cause de la caméra ou de changements de luminosité. Pour enlever ce bruit, des opérateurs morphologiques d’érosion et de dilatation sont appliqués afin d’effacer les pixels isolés.

L’image obtenue à cette étape de l’algorithme nous montre les contours des objets qui sont en mouvement dans la scène. Comme le but est de rechercher des objets en mouvement

et ayant des côtés droits, un algorithme de détection de ligne est appliqué sur cette image. La transformée de Hough (Duda et Hart, 1972) est donc utilisée pour trouver la ligne dominante dans l'image. Si cette ligne satisfait certains critères d'angle, l'algorithme conclut alors que l'objet recherché a été détecté.

L'article présente des résultats pour la détection de chariots et pour la détection du tiroir. Puisque la détection du chariot s'effectuait dans une zone où la luminosité changeait facilement, les résultats obtenus étaient beaucoup moins bons que ceux du tiroir. Le taux de détection du tiroir se situait à 96%. Le 4% d'erreur était principalement dû au fait que la tête du caissier causait parfois une occlusion du tiroir, ce qui empêchait sa détection. Ces résultats nous encouragent donc à utiliser une technique similaire lors de la détection des tiroirs pour notre projet.

Le dernier article Sturm *et al.* (2010) s'intéresse au problème des robots domestiques. En particulier, le problème d'ouverture des tiroirs et de portes par ces robots est examiné afin de trouver une méthode et d'en évaluer l'efficacité. Cette méthode a l'avantage de fonctionner sans l'utilisation d'aucun marqueur.

Comme les robots domestiques doivent effectuer des manipulations d'objets, il est souvent nécessaire qu'ils soient équipés d'une caméra de profondeur. Plusieurs techniques et systèmes de vision stéréo sont possibles afin d'obtenir des images de profondeur. Dans leur cas, Sturm *et al.* (2010) ont décidé de concevoir eux-mêmes leur propre système de vision stéréo. Leur système fonctionne à l'aide d'une texture qui est projetée sur la scène. Cette texture est ensuite filmée par deux caméras distinctes, ce qui leur permet d'obtenir la carte de disparités. Enfin, ce système leur permet d'avoir un taux de 15 images de profondeur par seconde.

À partir de l'image de profondeur, la première étape de leur méthode est de détecter les plans présents dans la scène, puisque la façade avant d'un tiroir ou d'une porte apparaît toujours comme un plan. Pour faire cette détection, Sturm *et al.* (2010) utilisent la méthode RANSAC (RANdom SAMple Consensus), Fischler et Bolles (1981). Cette méthode itérative permet de trouver les paramètres du plan qui représente le mieux un ensemble donné de points.

Une fois que les paramètres d'un plan sont trouvés, l'étape suivante est d'identifier, dans l'espace 3D, le rectangle qui coïncide le mieux avec les points de ce plan. Une méthode itérative tente de déterminer les paramètres de ce rectangle. Cette méthode définit d'abord une fonction de coût calculée selon l'ensemble des paramètres du rectangle. À chaque itération, cette fonction de coût est évaluée et l'ensemble de paramètres donnant le meilleur coût est conservé. La première itération débute avec un rectangle aléatoire à l'intérieur du plan. Le seul problème de cette méthode est qu'elle peut parfois converger vers un maximum local.

Pour éviter ce problème, l'algorithme est appliqué plusieurs fois en choisissant des rectangles différents à la première itération.

Pour enseigner au robot le mouvement d'ouverture d'un tiroir ou d'une porte en particulier, son mouvement d'ouverture est effectué devant ses caméras. Ensuite, à chaque trame de la séquence vidéo obtenue, les rectangles correspondants à la façade sont détectés par la méthode décrite précédemment. Par la suite, l'algorithme tente d'analyser ces différents rectangles pour en déduire une séquence de rectangles correspondant au mouvement d'ouverture du tiroir ou de la porte. Les paramètres de ce mouvement, décrit en termes de translation et de rotation, peuvent ainsi être calculés pour que le robot puisse reproduire le même mouvement d'ouverture.

Sturm *et al.* (2010) ont effectué des expériences avec différents types d'armoires de cuisine. Les résultats indiquent que dans plus de 75% des cas, la façade du tiroir était correctement détectée. D'autres résultats sont obtenus concernant la manipulation de ces tiroirs par le robot. Ces derniers résultats sont moins importants dans notre cas, puisque nous nous intéressons surtout à la détection de tiroirs, et non à leur ouverture.

Enfin, la méthode proposée par cet article est utile dans le cas où les tiroirs seraient vus de face par une caméra. Par contre, dans notre cas, les tiroirs sont toujours observés de haut et il serait donc difficile d'appliquer cette méthode. La méthode pourrait cependant nous donner certaines idées dans le cas où nous envisagerions d'utiliser des images de profondeur dans notre projet.

2.5 Résumé de la revue de littérature

Pour nous retrouver dans notre revue de littérature, nous pouvons récapituler les différents articles présentés sous la forme d'un tableau. Le tableau 2.1 résume chacun des 8 articles de la revue, en spécifiant le contexte du projet, le type de caméra utilisé, ainsi que le sujet filmé par la caméra. La dernière colonne indique le taux de bonnes détections obtenu selon les expériences de chaque projet. Certains projets, comme les brevets, ne comportaient aucun résultat.

On peut voir avec ce tableau que le contexte était toujours de type commercial ou de type domestique. En effet, on retrouve souvent des inventaires à gérer dans ces deux contextes. On pourrait aussi imaginer des projets se déroulant dans des contextes industriels, par exemple dans des entrepôts. Le fait que la caméra soit fixe ou mobile peut influencer le choix des algorithmes utilisés. Par exemple, si la caméra est mobile, il devient difficile d'appliquer un algorithme de soustraction d'arrière-plan. Par contre, une caméra mobile permet de surveiller

Tableau 2.1 Tableau résumant la revue de littérature

Auteur(s) et date	Contexte	Caméra	Sujet(s)	Résultats
Mankodiya <i>et al.</i> (2012)	commercial	mobile	inventaire	aucun
Francis <i>et al.</i> (2013)	commercial	mobile	inventaire	75%
Schwartz (2007)	commercial	fixe	inventaire	aucun
Opalach <i>et al.</i> (2012)	commercial	fixe	inventaire	aucun
Färnström <i>et al.</i> (2002)	domestique	fixe	denrées alimentaires	aucun
Hannan <i>et al.</i> (2012)	domestique	mobile	ordures ménagères	95%
Mustafa et Sethi (2005)	commercial	fixe	tiroir de caisse enregistreuse	96%
Sturm <i>et al.</i> (2010)	domestique	mobile	tiroirs et armoires de cuisine	75%

à elle seule un inventaire au complet, puisqu'elle peut être déplacée vers n'importe quel produit.

Enfin, seulement la moitié des articles présentaient des résultats. De plus, il faut noter que ces résultats ont été obtenus en laboratoire et non dans des conditions d'utilisation réelle, ou bien à partir d'une banque de vidéos créées par les chercheurs.

CHAPITRE 3

PREMIER THÈME - DÉTECTION DE TIROIR

La détection de tiroir est la première étape de notre méthodologie. Cette étape est nécessaire avant de pouvoir établir l'état de l'inventaire. Plusieurs approches sont envisagées pour repérer les tiroirs dans les images.

Une première approche est d'utiliser des marqueurs de couleurs distinctes qui sont apposés sur chacun des tiroirs. Cette approche est décrite dans les sections 3.2 et 3.3. Une deuxième approche est d'utiliser des algorithmes de détection de lignes afin de reconnaître les côtés du tiroir. Cette approche est décrite dans les sections 3.4 et 3.5.

Avant de pouvoir appliquer ces méthodes de détection de tiroir, il peut être nécessaire de corriger les distorsions causées par la caméra. Ces modèles de correction sont d'abord présentés dans la section 3.1. Ensuite, nous expliquons, chacune des deux approches de détection de tiroir pour ensuite les évaluer et les comparer dans le chapitre 6.

3.1 Correction de la distorsion

3.1.1 Choix de la caméra

Selon la caméra utilisée, il se peut que des distorsions optiques soient créées dans l'image. Le choix de la caméra est un aspect important du projet à considérer. Les critères d'une caméra à considérer sont entre autres la résolution d'image, le taux de trames par seconde et la largeur du champ de vision. Il peut en effet être avantageux de choisir une caméra offrant un angle de vue plus large. Une caméra avec un objectif grand-angle peut être placée plus près des tiroirs, sans que les tiroirs ne débordent en dehors de l'image. Si le champ de vision de la caméra est plus étroit, alors la caméra devra être placée un peu plus haut, afin de bien pouvoir voir les premiers tiroirs en haut du cabinet. En visitant quelques réserves d'hôpital, nous avons remarqué que le plafond peut souvent être plus bas qu'un plafond normal. Avec ce type de plafonds, une caméra ayant un champ de vision normal ne pourrait pas voir en entier les premiers tiroirs. Dans ce cas, il faut absolument choisir une caméra à objectif grand-angle.

3.1.2 Distorsion radiale

Les caméras à objectif grand-angle ont par contre l'inconvénient de déformer l'image. Bien qu'il existe plusieurs types de distorsions, le type de distorsion le plus fréquent est la

distorsion radiale. Parmi les distorsions radiales, on retrouve les distorsions en barillet et les distorsions en coussinet. Dans notre cas, la caméra que nous avons choisie présentait des images avec une distorsion de type barillet tel que montrée dans la figure 3.1 Cette distorsion est problématique puisque les algorithmes de détection de tiroir que nous utilisons supposent tous que les projections de la caméra sont linéaires, c'est-à-dire que les lignes du monde réel deviennent des lignes droites dans l'image. Les lignes des tiroirs rectangulaires sont cependant déformées, et il faut préalablement procéder à une correction de l'image. Les distorsions radiales peuvent heureusement être corrigées afin d'obtenir une image normale comme à la figure 3.3.

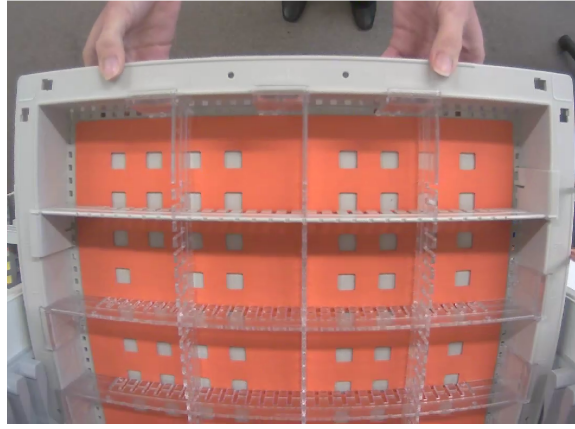


Figure 3.1 Image de la caméra avec distorsion en barillet

La distorsion radiale est due au rayon de courbure des lentilles. On peut observer sur l'image déformée que la distorsion est faible au centre, mais qu'elle s'accroît en bordure de l'image. En effet, le déplacement d'un point de l'image est proportionnel à sa distance radiale avec le centre de l'image. Les équations 3.1 présentent un modèle de distorsion radiale. Ce modèle utilise les premiers termes d'une série de Taylor autour de $r = 0$, r étant la distance entre un point donné et le centre.

$$\begin{aligned}
 x_{corr} &= (x_i - x_c)(1 + K_1 r^2 + K_2 r^4 + K_3 r^6 + \dots) \\
 y_{corr} &= (y_i - y_c)(1 + K_1 r^2 + K_2 r^4 + K_3 r^6 + \dots) \\
 r &= \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2}
 \end{aligned} \tag{3.1}$$

où (x_i, y_i) est le point initial dans l'image déformée, (x_{corr}, y_{corr}) est la nouvelle position du point sur l'image corrigée, r est la distance du point initial par rapport au centre de distorsion (x_c, y_c) , et K_1 , K_2 et K_3 sont les paramètres de distorsion radiale.

Normalement, dans la plupart des cas, il est suffisant d'utiliser un modèle qui contient seulement les deux premiers termes de la série de Taylor. Par contre, si la caméra est de moins bonne qualité, il devient nécessaire d'ajouter un troisième terme au modèle avec un autre coefficient K_3 . Dans notre cas, on peut tenter d'utiliser seulement les deux premiers coefficients K_1 et K_2 pour obtenir l'image de la figure 3.2. Cependant, en observant le résultat de cette correction, on remarque que les trois côtés du tiroir ne sont pas parfaitement linéaires, mais qu'ils sont légèrement bombés. Cette courbure est encore plus visible dans les deux coins inférieurs de l'image. Les séparateurs se trouvant près du centre de l'image, quant à eux, sont plus droits parce que la distorsion radiale est plus faible dans cette région de l'image. Pour avoir une correction plus précise, on préfère donc ajouter un troisième coefficient K_3 comme dans l'image 3.3.

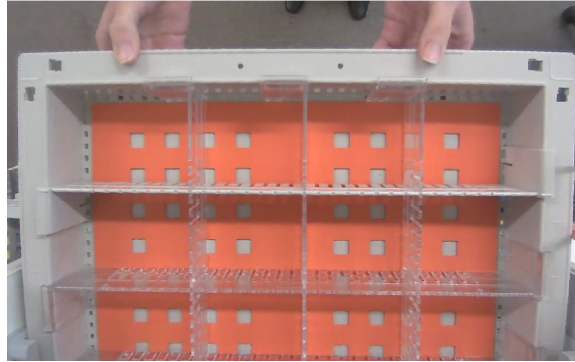


Figure 3.2 Image de la caméra avec correction de la distorsion radiale avec K_1 et K_2

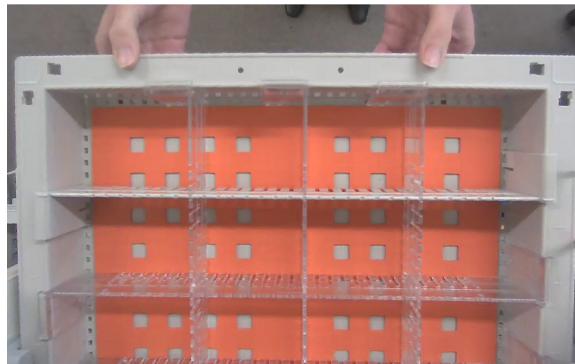


Figure 3.3 Image de la caméra avec correction de la distorsion radiale avec K_1 , K_2 et K_3

3.1.3 Distorsion tangentielle

Le modèle de distorsion radiale ne corrige pas toujours complètement toutes les distorsions. Parfois, avec une caméra de moins bonne qualité, des déformations peuvent rester présentes dans l'image même après une correction de la distorsion radiale.

Pour corriger complètement l'image, il faut alors tenir compte d'un deuxième type de distorsion. Il s'agit de la distorsion tangentielle. Cette distorsion est due au fait qu'il est difficile d'assembler la lentille dans la caméra pour qu'elle soit parfaitement alignée avec le plan du film ou du capteur. Pour corriger ce défaut, il faut ajouter deux autres termes polynomiaux aux équations de distorsion radiale. Ceci vient compléter nos équations et ajoute deux nouveaux coefficients, P_1 et P_2 .

$$\begin{aligned} x_{corr} &= (x_i - x_c)(1 + K_1 r^2 + K_2 r^4 + \dots) + 2P_1(x_i - x_c)(y_i - y_c) + P_2(r^2 + 2(x_i - x_c)^2) \\ y_{corr} &= (y_i - y_c)(1 + K_1 r^2 + K_2 r^4 + \dots) + P_1(r^2 + 2(y_i - y_c)^2) + 2P_2(x_i - x_c)(y_i - y_c) \\ r &= \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} \end{aligned} \tag{3.2}$$

où P_1 et P_2 sont les paramètres de distorsion tangentielle.

3.1.4 Estimation des paramètres de distorsion

Il existe plusieurs techniques pour estimer les différents paramètres de distorsion. Une de ces techniques est de photographier des scènes contenant beaucoup de lignes droites, surtout en périphérie de l'image. Ensuite, on cherche à ajuster les paramètres jusqu'à ce que toutes ces lignes deviennent complètement droites. Cette méthode, nommé "plumb-line method" ou "plumb-line constraint", est utilisée plusieurs fois dans la littérature.

Une autre approche est d'utiliser un objet de calibrage de forme connue comme un échiquier d'où l'on peut déduire les coordonnées 3D de points caractéristiques. Ensuite, les paramètres intrinsèques et extrinsèques de la caméra peuvent être estimés en même temps que les paramètres de distorsion radiale et tangentielle. C'est cette approche que nous avons suivie pour estimer la valeur de nos paramètres.

En appliquant l'approche précédente, nous obtenons des valeurs de $K_1 = -4.78 \times 10^{-7}$, $K_2 = 2.79 \times 10^{-13}$, $K_3 = -9.37 \times 10^{-20}$ et $(x_c, y_c) = (640, 330)$ avec des images faisant 1280 par 800 pixels. Nous obtenons aussi des valeurs de $P_1 = 5.07 \times 10^{-9}$, $P_2 = 4.84 \times 10^{-9}$ comme paramètres de distorsion tangentielle. Par contre, comme ces valeurs de P_1 et P_2 sont très faibles et qu'ils ont très peu d'effet sur le résultat de la correction, on pourrait considérer que la caméra ne présente pas de distorsion tangentielle. La correction de distorsion radiale avec les

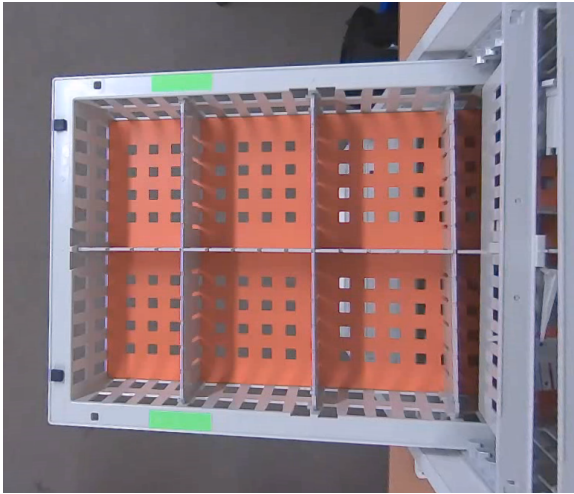
valeurs de K_1 , K_2 et K_3 nous permet d'obtenir des images corrigées de façon satisfaisante. Maintenant, les lignes de nos images sont suffisamment droites pour éventuellement être détectées par nos algorithmes de détection de tiroir.

3.2 Détection de marqueurs de tiroir

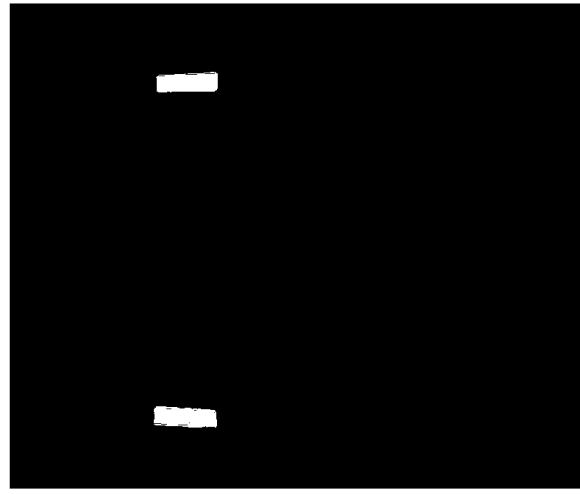
Un aspect intéressant de ce projet de vision artificielle est que nous avons un certain contrôle sur la configuration et la disposition de l'inventaire. L'hôpital et la compagnie Logi-D permettent de réarranger les cabinets d'inventaire, au besoin, si cela permet de faciliter la détection.

Dans le domaine de la vision, il est assez courant d'utiliser des marqueurs de couleurs distinctes afin de pouvoir segmenter plus facilement des objets dans un environnement. On veut donc répéter cette approche ici : l'utilisation de marqueurs permettra de reconnaître les tiroirs lorsqu'ils sont ouverts.

La première méthode décrite ici consiste donc à utiliser des marqueurs. Des marqueurs de couleurs sont posés sur chacun des deux côtés de chaque tiroir, vers l'avant, comme dans la figure 3.4(a).



(a) Positionnement des marqueurs sur un tiroir.



(b) Résultat de la segmentation des marqueurs.

Figure 3.4 Positions et segmentations des marqueurs verts fluorescents.

3.2.1 Forme des marqueurs

La forme rectangulaire des marqueurs permettra aussi de déduire l'orientation du marqueur et du tiroir : la direction du mouvement du tiroir s'obtiendra en prolongeant, d'une

ligne imaginaire, le marqueur dans le sens de sa longueur. Aussi, avant de conclure qu'un tiroir est détecté, l'algorithme pourra vérifier que la paire de marqueurs détectée contient bien deux marqueurs ayant des orientations semblables.

3.2.2 Segmentation de marqueurs

La détection de marqueurs est grandement facilitée si leur couleur est bien distincte des autres couleurs pouvant être présentes dans l'image. Par exemple, pour le projet de Logi-D, une couleur de marqueur vert fluorescent a été choisie parce que peu d'items médicaux portent cette couleur. Le vert fluorescent est en effet une couleur peu fréquente.

Une fois les marqueurs bien posés sur chaque tiroir, reconnaître les tiroirs revient principalement à résoudre un problème de segmentation. Pour segmenter le vert des marqueurs, des tests sont d'abord effectués pour choisir l'espace de couleurs le plus approprié, qui peut être un autre espace que l'espace RGB. Il est connu que le choix d'un bon espace de couleur peut beaucoup faciliter la segmentation.

En essayant plusieurs espaces de couleur différents, on observe que l'espace YCbCr permet de segmenter plus facilement les marqueurs verts. L'espace YCbCr s'adapte bien à notre situation parce qu'il permet de faire une séparation entre les informations de chrominances (Cb et Cr) et l'information de la luminance (Y). Les formules qui permettent de passer de l'espace RGB à l'espace YCbCr sont montrées par les équations 3.3. On voit que la composante de luminance (Y) correspond en fait à la conversion d'une image couleur en niveau de gris. La composante Cb indique la quantité de bleu relativement à la luminance, tandis que la composante Cr indique la quantité de rouge par rapport à la luminance. La quantité de vert peut être déduite en connaissant la luminance (Y) et les chrominances rouge (Cr) et bleu (Cb). On ajoute 128 aux valeurs de Cb et Cr pour éviter d'avoir des valeurs négatives.

Pour détecter des marqueurs verts, il faudra donc chercher les pixels où les chrominances Cb et Cr sont faibles. On peut voir l'effet des chrominances Cb et Cr lorsque la luminance reste constante dans la figure 3.5.

$$\begin{aligned}
 Y &= 0.299 * R + 0.587 * G + 0.114 * B \\
 Cb &= (B - Y) * 0.564 + 128 \\
 Cr &= (R - Y) * 0.713 + 128
 \end{aligned}
 \tag{3.3}$$

Avec cet espace de couleur, la segmentation s'effectue en appliquant un seuillage sur les valeurs de chaque pixel de l'image. L'image obtenue (figure 3.4(b)) après ce premier seuillage

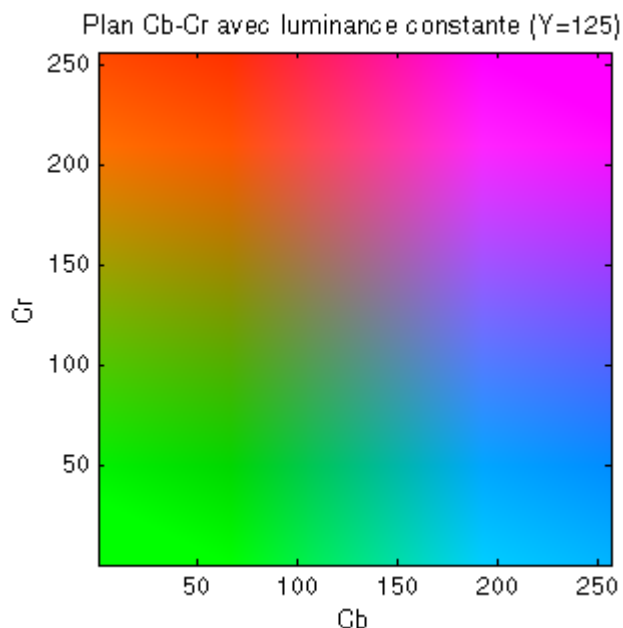


Figure 3.5 Composantes Cb et Cr de l'espace YCbCr.

est une image binaire où les pixels blancs correspondent à des pixels de marqueurs. Notons que les seuils de segmentation utilisés doivent être choisis durant une procédure de calibration, puisque ces valeurs varient beaucoup avec les facteurs d'éclairage de la pièce.

Avec l'image binaire obtenue, il faut ensuite identifier les groupes de pixels blancs pouvant correspondre à des marqueurs. Il s'agit de reconnaître les pixels qui sont connectés entre eux et qui forment des groupes de tailles suffisamment grandes (un seuillage sur la taille est effectué ici). Dans Matlab par exemple, la fonction `regionprops()` permet d'obtenir des statistiques sur chacun des regroupements de pixels, comme la taille, l'orientation principale de la forme, etc.

3.2.3 Identification des paires de marqueurs

La prochaine étape consiste à identifier, dans l'image binaire obtenue (figure 3.4(b)), une paire de marqueurs. On ajoute quelques contraintes sur la détection de cette paire pour s'assurer qu'il n'y a pas de fausse détection :

1. L'aire de chaque marqueur doit être située à l'intérieur d'un intervalle d'aires acceptables, définie durant la calibration du système.
2. Les aires des deux marqueurs doivent être semblables, c'est-à-dire que le rapport des deux aires doit être suffisamment près de 1, selon un certain seuil.

3. La distance horizontale, selon l'axe des X, entre le barycentre de chaque marqueur doit être inférieure à un certain seuil.
4. Les marqueurs doivent être symétriques selon l'axe horizontal de l'image, c'est-à-dire que le point milieu des deux barycentres des marqueurs doit se trouver suffisamment près du centre vertical de l'image, selon un certain seuil. Ceci implique que la caméra soit bien alignée.
5. La distance entre les deux marqueurs doit correspondre à une des distances enregistrées pour un des tiroirs du cabinet pendant la calibration. Le rapport de la distance détectée sur la distance enregistrée doit être suffisamment près de 1, selon un certain seuil.

Il faut noter que tous ces seuils peuvent être définis durant la phase de calibration du système. On ajoute aussi deux autres contraintes qui sont évaluées en faisant correspondre une ellipse à chacun des marqueurs. L'ellipse correspondante est trouvée en calculant les moments d'ordre 2 de la région du marqueur. On définit ensuite une ellipse ayant ces mêmes moments. Les caractéristiques des ellipses nous permettent d'ajouter les deux autres contraintes suivantes :

6. La forme de l'ellipse correspondant à chaque marqueur doit être suffisamment allongée, c'est-à-dire que l'excentricité de l'ellipse doit être plus grande qu'un certain seuil.
7. L'orientation du grand axe de l'ellipse correspondante doit être suffisamment à horizontale, c'est-à-dire à l'intérieur d'un intervalle en degrés.

Finalement, lorsqu'une paire de marqueurs respectant toutes ces contraintes est détectée, on en conclut qu'un tiroir ouvert est présent dans l'image. La position du tiroir peut aussi être déduite à partir de la position des marqueurs, puisque les marqueurs sont toujours positionnés au même endroit sur les tiroirs.

3.2.4 Limitations de la détection de marqueurs

Malgré tout, l'algorithme de détection de marqueurs peut comporter certaines limitations que nous présentons ici. D'abord, les marqueurs doivent être posés soigneusement à une position précise sur chacun des tiroirs. Ceci peut demander un certain temps, surtout si les cabinets sont nombreux, puisqu'on peut facilement retrouver des centaines de cabinets à l'intérieur d'un seul hôpital.

Ensuite, les marqueurs peuvent parfois être cachés, soit par la main ou la tête d'un utilisateur qui se penche au-dessus du tiroir, ou bien par un item qui déborde sur le côté

du tiroir. Ce dernier cas peut cependant être évité en utilisant des tiroirs plus profonds pour éviter d’avoir des items qui débordent.

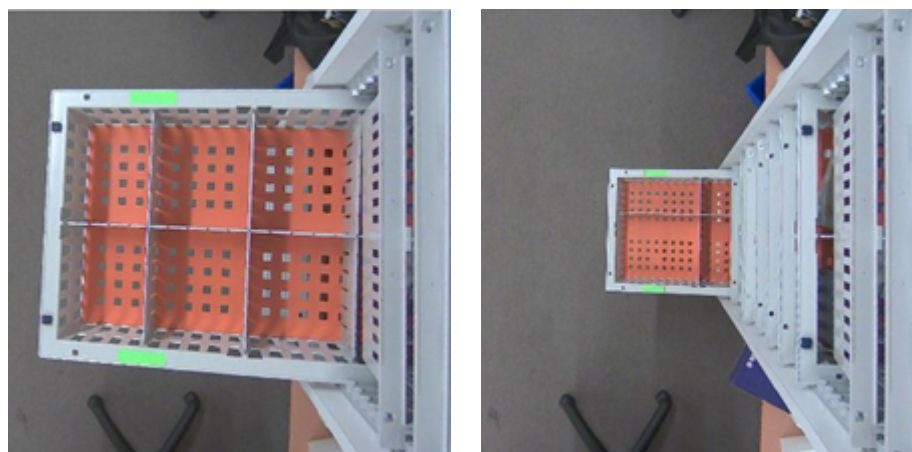
Enfin, il se peut même que les marqueurs soient enlevés par l’hôpital pour des raisons de salubrité. C’est justement ce qui s’est produit à l’Hôpital du Sacré-Cœur de Montréal : le service de salubrité a dû effectuer un nettoyage pour éviter une contamination des patients, mais en même temps, des marqueurs ont été enlevés.

3.3 Identification de tiroir par les marqueurs

3.3.1 Choix de la méthode d’identification

Une fois qu’un tiroir est reconnu par le système, il faut aussi pouvoir identifier de quel tiroir il s’agit afin de rendre possible la surveillance d’inventaire. Les cabinets de Logi-D, utilisés dans notre projet, comportent normalement une dizaine de tiroirs chacun. Comme la caméra est positionnée de façon fixe au plafond, la distance entre un tiroir donné et la caméra reste toujours la même. En observant les tiroirs du point de vue de la caméra dans la figure 3.6, on voit qu’un effet de perspective important se présente : les tiroirs du bas du cabinet paraissent beaucoup plus petits que les tiroirs du haut. Une autre conséquence de la perspective est que la distance entre les marqueurs d’un tiroir bas sera plus petite que pour un tiroir plus haut. L’idée pour identifier un tiroir est donc d’utiliser comme identifiant cette distance entre la paire de marqueurs. Des mesures en pixels des distances entre chaque paire de marqueurs doivent être prises au préalable lors d’une procédure de calibration. Ensuite, lorsqu’une paire de marqueurs sera détectée pendant l’ouverture d’un tiroir, il suffira de mesurer la distance en pixels séparant les marqueurs et de trouver la distance correspondante parmi les distances mesurées à la calibration.

Une autre approche envisageable pour identifier les tiroirs aurait été d’utiliser un certain type de code-barre à deux-dimensions, comme le code QR par exemple. Le problème avec cette approche est que la caméra utilisée n’offre pas toujours une résolution d’image assez grande pour pouvoir lire distinctement le code-barre du dernier tiroir. Aussi, cette approche aurait nécessité d’ajouter d’autres autocollants aux tiroirs en plus des marqueurs déjà présents. C’est pour cela que la première approche, celle utilisant les distances entre les paires de marqueurs, semble être plus appropriée au problème. De cette façon, les marqueurs servent non seulement à trouver la position d’un tiroir dans l’image, mais aussi à identifier de quel tiroir il s’agit.



(a) Tiroir en haut du cabinet

(b) Tiroir en bas du cabinet

Figure 3.6 Effet de perspective de la caméra

3.3.2 Limitations

La principale limitation de cette méthode d'identification de tiroir se produit lorsque les tiroirs sont inclinés comme dans la figure 3.7. En effet, avec un tiroir incliné vers l'utilisateur, plus on ouvre le tiroir, plus les marqueurs s'éloignent de la caméra. Il devient donc difficile d'identifier ces tiroirs avec une seule valeur de distance, parce que la distance devient variable selon l'ouverture du tiroir.

Heureusement, les tiroirs inclinés sont normalement beaucoup moins nombreux que les tiroirs droits. Il y en a seulement un ou deux par cabinets. Le plus simple, pour contrer cette limitation, serait de réorganiser les rangements des cabinets pour avoir uniquement des tiroirs droits. Le seul inconvénient à cette réorganisation est que les cabinets perdent alors un peu de leur espace de rangement dans leur partie supérieure.

3.4 Détection de tiroir par les côtés

Comme nous l'avons vu dans la section 3.2, les marqueurs sont utiles parce qu'ils permettent de faciliter l'algorithme de détection. Par contre, l'utilisation de marqueurs comporte certaines limitations comme le temps de pose des marqueurs et l'occlusion de ceux-ci par l'utilisateur. Dans cette deuxième méthode de détection de tiroir, nous considérons la détection de tiroir sans marqueur.



Figure 3.7 Un exemple de tiroir incliné.

3.4.1 Caractéristiques des tiroirs

Pour concevoir un algorithme qui peut reconnaître les tiroirs sans l'usage de marqueurs, on peut d'abord réfléchir à ce qui caractérise physiquement un tiroir. Du point de vue de la caméra placée en haut du cabinet, les tiroirs apparaissent comme des rectangles légèrement déformés par la perspective, puisque la caméra est placée de façon à pointer directement vers le sol. De plus, le mouvement d'ouverture et de fermeture des tiroirs se fait toujours selon le même axe. Nous avons donc deux caractéristiques importantes à notre disposition pour pouvoir identifier les tiroirs. La première caractéristique est que ce sont des objets en mouvement lorsqu'ils sont utilisés. La deuxième caractéristique est qu'ils sont formés de côtés droits que l'on peut considérer comme parallèles entre eux, puisque nos expériences montrent que, malgré l'effet de perspective, l'angle entre deux côtés opposés du tiroir ne dépasse jamais 10 degrés.

Dans la littérature, il existe déjà des algorithmes de traitement d'images permettant de reconnaître ces deux caractéristiques. Pour faire la détection du mouvement, il existe plusieurs algorithmes de soustraction d'arrière-plan, dont celle de Mckenna *et al.* (2000). Quant à la détection des côtés de tiroir, il existe des algorithmes de détection de lignes comme la transformée de Hough (Hough, 1962). Nous proposons donc de combiner ces deux

algorithmes, afin de pouvoir détecter les lignes parallèles qui sont en mouvement et qui correspondent probablement à celles du tiroir.

3.4.2 Soustraction d'arrière-plan

La première étape de notre algorithme est de détecter les objets qui sont en mouvement. Pour cela, nous utilisons directement la soustraction d'arrière-plan de McKenna *et al.* (2000). Dans sa forme la plus simple, une soustraction d'arrière-plan consiste à soustraire l'image actuelle de l'image de référence comme dans l'équation 3.4. L'image de référence est une image de la scène où tous les objets sont fixes. Le résultat de cette soustraction nous donne les changements pour chacun des pixels de l'image. On peut ensuite appliquer un seuil donné sur cette image pour obtenir une image binaire où les pixels blancs correspondent à des pixels de mouvement comme dans la figure 3.8.

$$|I_{actuelle} - I_{reference}| > seuil \quad (3.4)$$

Avec cette soustraction d'arrière-plan, plusieurs suppositions sont faites. D'abord, on suppose que la caméra reste fixe, mais aussi que l'arrière-plan reste statique. Une autre supposition est que l'éclairage reste toujours constant. S'il y a des changements de luminosité dans la scène, alors l'image de référence devient erronée et il faut la réinitialiser. Dans notre cas, il peut arriver que les cabinets soient placés près d'une fenêtre. Ceci vient alors créer des changements d'éclairage à notre image de référence tout au cours de la journée, ou même en un court intervalle de temps, si le soleil devient caché par des nuages, par exemple. La dernière supposition d'un éclairage constant est, de toutes nos suppositions, celle qui risque d'être la moins respectée.

La méthode de McKenna vise justement à éviter de devoir supposer que l'éclairage reste constant. Avec cette méthode, l'image de référence est mise à jour continuellement, ce qui permet de régler les problèmes de changement d'illumination. Plutôt que d'enregistrer une seule image de référence fixe, l'algorithme de McKenna enregistre, pour chaque pixel, sa moyenne et sa variance. Ces valeurs de moyenne et de variance sont mises à jour à chaque trame, pourvu que le pixel à mettre à jour corresponde bien à un pixel d'arrière-plan. Les équations de mise à jour de la variance dépendent d'un seul paramètre, α . Ce paramètre α se situe entre 0 et 1, et sert à déterminer à quelle vitesse l'image de référence peut changer. Avec un paramètre α élevé, l'image de référence change lentement alors qu'un paramètre α bas permet de faire changer l'image de référence plus rapidement. L'équation 3.5 indique comment mettre à jour la moyenne pour chaque pixel de l'image de référence.

$$\mu_{t+1} = \alpha\mu_t + (1 - \alpha)z_{t+1} \quad (3.5)$$

où μ_t et μ_{t+1} sont les moyennes du pixel aux temps t et $t+1$, α est le taux d'adaptation de l'image de référence, z_{t+1} , est la mesure du pixel au temps $t+1$.

Notons que la valeur de la constante α est choisie de façon empirique puisqu'elle dépend à la fois du nombre de trames par seconde et de la vitesse à laquelle la scène change. Dans notre cas, nous avons choisi une valeur de 0.9. Quant à la valeur initiale de μ au temps $t=0$, il est préférable d'utiliser une trame contenant seulement l'arrière-plan.

L'équation 3.6 de mise à jour de la variance est semblable à celle pour la moyenne : le même paramètre α est également utilisé.

$$\sigma_{t+1}^2 = \alpha(\sigma_t^2 + (\mu_{t+1} - \mu_t)^2) + (1 - \alpha)(z_{t+1} - \mu_{t+1})^2 \quad (3.6)$$

où σ_t^2 et σ_{t+1}^2 sont les variances du pixel aux temps t et $t+1$.

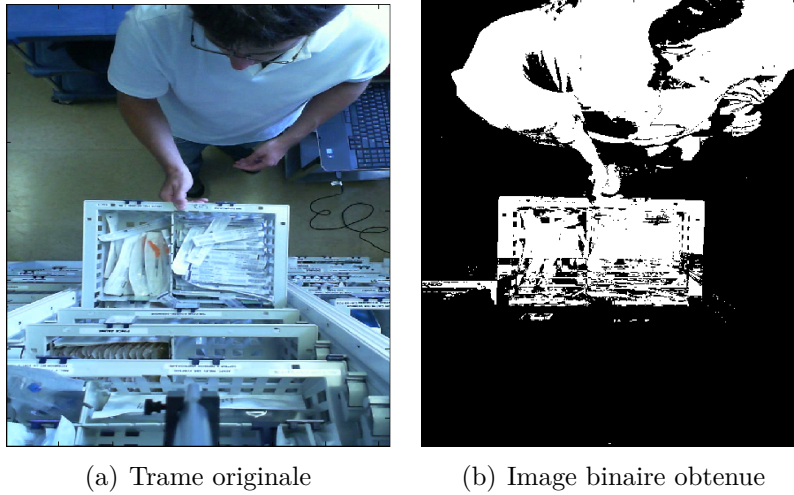


Figure 3.8 Exemple de soustraction d'arrière-plan

3.4.3 Détection de lignes

Une fois que l'arrière-plan a été soustrait, on obtient alors l'image binaire de soustraction d'arrière-plan (figure 3.8). Cette image est en fait une segmentation de l'arrière-plan et de l'avant-plan, c'est-à-dire que tous les objets en mouvement dans l'image sont blancs. Dans la figure 3.8, les objets mobiles que nous pouvons observer sont le tiroir en bas de l'image, et l'utilisateur en haut. Parmi tous les objets possiblement en mouvement, le but est de pouvoir identifier la position du tiroir. Le tiroir est facilement reconnaissable par ses côtés droits. On

utilise donc un algorithme de détection de lignes afin de trouver dans l'image binaire une paire de lignes verticales, de longueurs semblables et ayant plus ou moins la même direction. On fixe un seuil en degrés pour déterminer si les lignes ont la même direction. Notre approche suppose que les tiroirs sont probablement les seuls objets à avoir des caractéristiques rectangulaires et à être en même temps mobiles.

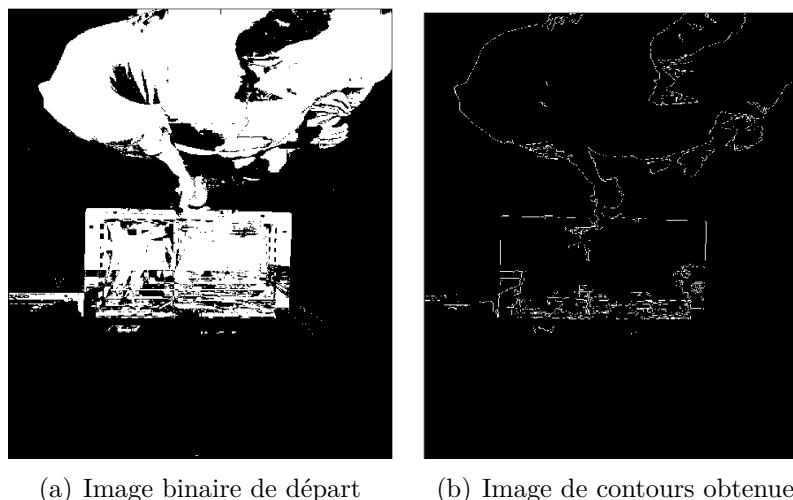


Figure 3.9 Exemple d'extraction des contours.

Avant de pouvoir extraire des lignes dans l'image, il faut d'abord avoir une image de contours. Ces contours s'obtiennent facilement à partir de l'image binaire (figure 3.9(a)). Plusieurs méthodes sont possibles pour avoir les contours d'une image : on peut faire une convolution avec un filtre de Sobel, ou encore utiliser l'algorithme de Canny (Canny, 1986). La figure 3.9(b) montre le résultat d'une extraction de contours sur l'image binaire de la figure 3.9(a). Dans ce cas, nous avons utilisé l'algorithme de détection de voisinage de Moore modifié par le critère d'arrêt de Jacob tel que décrit par Gonzalez *et al.* (2003). Cet algorithme est implémenté par la fonction `bwboundaries()` de Matlab. Cette fonction prend en entrée une image binaire comme la nôtre et retourne toutes les frontières de l'image. Il faut noter que l'algorithme de Moore peut uniquement être utilisé sur des images binaires, contrairement à l'algorithme de Sobel ou de Canny. Par contre, cet algorithme est beaucoup plus simple à utiliser parce qu'il ne demande aucun seuil ou paramètre.

Maintenant que l'image des contours est obtenue, un algorithme d'extraction de droites peut être appliqué. Nous utilisons ici l'algorithme classique de la transformée de Hough avec les modifications qui ont été apportées par Duda et Hart (1972) pour représenter les droites de façon paramétrique. Dans cet algorithme, les droites sont définies par deux paramètres : ρ

et θ . Plutôt que de définir les droites avec la forme $y_i = ax_i + b$, on utilise la forme normale donnée par l'équation 3.7 :

$$x_i \cos(\theta) + y_i \sin(\theta) = \rho \quad (3.7)$$

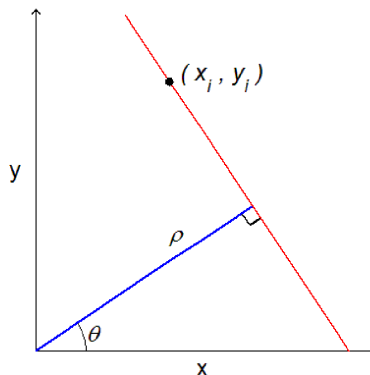


Figure 3.10 Les paramètres ρ et θ d'une droite

La figure 3.10 illustre comment les paramètres ρ et θ déterminent la géométrie d'une droite. Le paramètre ρ représente la distance entre la droite et l'origine, alors que le paramètre θ indique l'angle que fait la normale reliant l'origine et la droite. Avec ces deux paramètres ρ et θ , l'algorithme de Hough définit un espace à deux dimensions qu'on appelle l'espace de Hough. Cet espace sert à contenir toutes les droites possibles dans l'image. Par exemple, considérons la situation montrée à la figure 3.11 où nous voulons trouver la droite reliant les points A et B. On considère d'abord le point A situé à $(x=2\sqrt{3}, y=2)$ dans l'image. On sait qu'il y aura une infinité de droites pouvant passer par ce point. Pour passer par ce point A, les droites doivent satisfaire l'équation normale, $\rho = 2\sqrt{3} \cos(\theta) + 2 \sin(\theta)$, représentée en bleu dans la figure 3.12. Cette courbe dans l'espace de Hough nous permet ainsi de connaître les paramètres ρ et θ de toutes les droites passant par le point A.

L'algorithme trace la courbe correspondante dans l'espace de Hough pour chacun des points de l'image. Par exemple, avec un deuxième point B, on obtient une autre courbe dans l'espace de Hough qui décrit toutes les lignes possibles passant par ce point. Si l'on veut trouver la ligne passant à la fois par le point A et B, on peut regarder les 2 courbes tracées dans l'espace de Hough (figure 3.12) et trouver leur point d'intersection. Dans notre exemple, la solution peut être calculée en résolvant nos deux équations. On trouve que θ doit être de 30 degrés et que ρ doit valoir 4. Avec une image plus complexe et un plus grand nombre de points, l'espace de Hough contient alors une très grande quantité de courbes et il est difficile de trouver une solution de façon analytique. On utilise donc une méthode numérique.

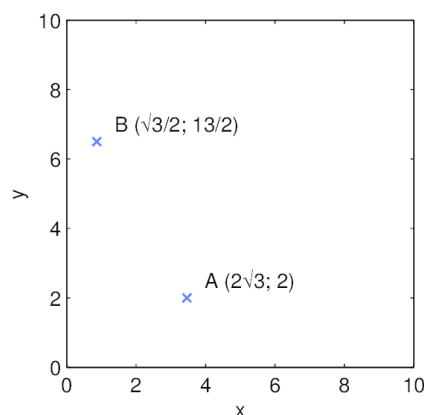


Figure 3.11 Deux points A et B dans une image

Pour résoudre le problème de façon numérique, on doit d'abord discrétiser l'espace de Hough. Pour cela, on utilise une grande matrice à deux dimensions, qui sert d'accumulateur, telle que montrée à la figure 3.13.

Cette matrice contient la même information que dans le graphique de la figure 3.12, mais représenté d'une autre façon. À gauche de la grande matrice, une matrice colonne sert à remplacer l'axe des ordonnées et nous indique les valeurs de ρ . La matrice ligne en bas remplace aussi l'axe des abscisses et nous indique les valeurs de θ en degrés. On y retrouve les deux mêmes courbes en rouge et en bleu que dans la figure 3.12. On peut aussi vérifier que l'intersection des 2 courbes se situe au même endroit, c'est-à-dire au point $\rho = 4$ et $\theta = 30$ degrés. La valeur à ce point de l'accumulateur vaut 2 parce que les deux courbes passent par ce point. Normalement, on utiliserait une matrice de taille beaucoup plus grande pour avoir une plus grande précision sur les résultats de l'algorithme.

On peut donc utiliser cette matrice plutôt que de tracer des courbes dans l'espace de Hough. Il suffit d'incrémenter de un tous les éléments de la matrice qui se trouvent sur le chemin de la courbe à tracer. De cette façon, la matrice sert d'accumulateur pour chaque courbe. On dit que la transformée de Hough est un algorithme de votes parce que chaque incrémentation dans la matrice équivaut à faire un vote.

Une fois que tous les points de l'image ont été traités par l'algorithme, l'accumulateur contient alors un grand ensemble de droites possibles. Les droites les plus importantes sont celles qui ont le plus de votes. Pour trouver ces droites, il suffit de prendre les valeurs maximales dans l'accumulateur.

Dans notre cas, nous utilisons l'algorithme de Hough uniquement pour retrouver les côtés des tiroirs. En supposant que la caméra est bien alignée avec les tiroirs, on sait que les droites à détecter sont nécessairement plus verticales qu'horizontales. Il serait improbable

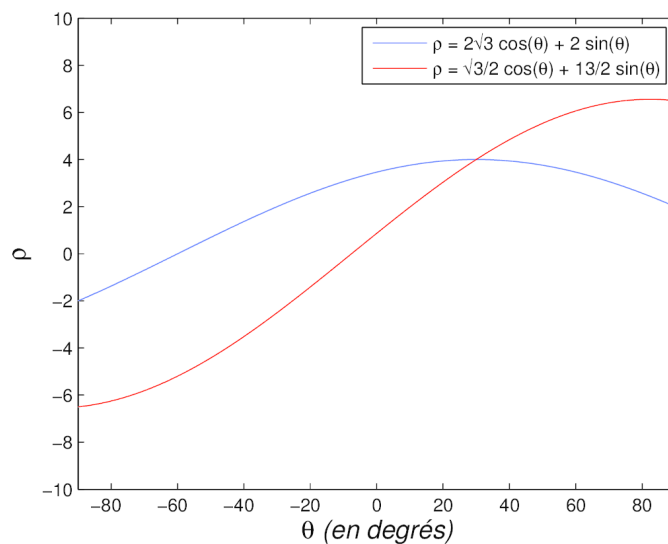
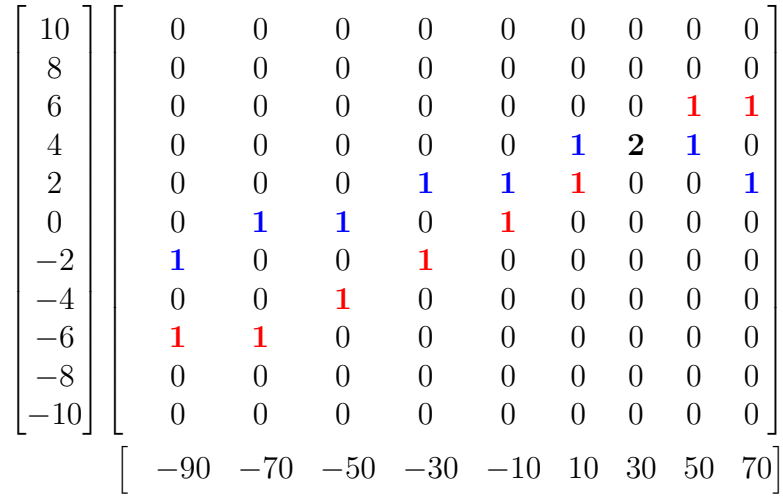


Figure 3.12 L'espace de Hough

que les lignes des tiroirs soient horizontales ou diagonales, à moins que la caméra soit mal alignée. Une façon d'améliorer la vitesse de nos calculs est donc de réduire la taille de la matrice qui sert d'accumulateur. Plutôt que de chercher les droites avec des angles allant de -90 degrés à +90 degrés comme dans la figure 3.13, on peut réduire notre recherche à un intervalle plus petit entre -10 degrés et +10 degrés par exemple. L'intervalle de -10 à 10 degrés est choisi ici de façon empirique, selon ce que nos expériences nous ont permis d'observer.

L'algorithme de Hough est déjà implémenté dans le logiciel Matlab. Dans notre cas, nous utilisons trois fonctions Matlab : `hough()`, `houghpeaks()` et `houghlines()`. La première fonction `hough()` prend une image binaire en entrée et retourne l'accumulateur dans une matrice. La deuxième fonction `houghpeaks()` identifie les sommets dans l'accumulateur et retourne leur position. Enfin, la troisième fonction `houghlines()` prend en paramètre les positions des sommets et retourne les caractéristiques des lignes correspondantes. La figure 3.14 indique les deux lignes verticales les plus importantes détectées par l'algorithme de Hough.

Comme on peut le constater dans la figure 3.14, notre détection de lignes par la transformée de Hough est utilisée seulement pour détecter les deux lignes verticales du tiroir. Nous avons aussi tenté d'appliquer cet algorithme pour détecter la bordure avant du tiroir. La détection de cette ligne horizontale serait utile afin de pouvoir localiser complètement le tiroir. Par contre, cette détection ne donne pas d'aussi bons résultats. Ceci s'explique par le fait que les utilisateurs cachent souvent cette partie du tiroir, que ce soit en ouvrant ou en fermant le tiroir, ou encore en se penchant au-dessus du tiroir. Avec ces manipulations, les utilisateurs viennent donc masquer la ligne qui devient alors invisible au système. En théorie,



$$\rho = 2\sqrt{3} \cos(\theta) + 2 \sin(\theta)$$

$$\rho = \frac{\sqrt{3}}{2} \cos(\theta) + \frac{13}{2} \sin(\theta)$$

Figure 3.13 Accumulateur pour l'algorithme de Hough

il serait possible d'informer le personnel de ne pas obstruer le devant du tiroir. Par contre, en pratique, il y aura toujours des cas où des utilisateurs pressés cachent le tiroir, surtout lorsque des soins d'urgence doivent être administrés rapidement.

3.5 Identification de tiroir par les côtés

Dans la section précédente, nous avons vu comment un tiroir pouvait être détecté par ses côtés. Une fois qu'un tiroir est détecté, le système doit aussi savoir de quel tiroir il s'agit afin d'établir une correspondance avec les produits de l'inventaire. Nous avons vu comment un tiroir pouvait être identifié à l'aide de marqueurs dans la section 3.3. Dans cette section-ci, nous nous intéressons à une méthode d'identification sans marqueur.

Comme dans la méthode avec marqueurs, nous utilisons ici le fait que les tiroirs en bas du cabinet apparaissent plus petits du point de vue de la caméra. Ainsi, les deux lignes verticales du tiroir, qui ont été détectées à la section précédente, apparaîtront plus rapprochées pour un tiroir bas et plus éloignées pour un tiroir haut. Cependant, on ne peut pas obtenir une mesure de distance fixe entre deux lignes si elles ne sont pas parfaitement parallèles. Une solution possible est de mesurer la distance entre ces lignes là où elles croisent les côtés du cabinet tel que montré dans la figure 3.15. Les lignes rouges indiquant les côtés du cabinet peuvent être obtenues pendant une phase de calibration lorsque le système est installé. Pendant cette calibration, le système doit aussi enregistrer, pour chaque tiroir, les distances entre les deux



Figure 3.14 Transformée de Hough

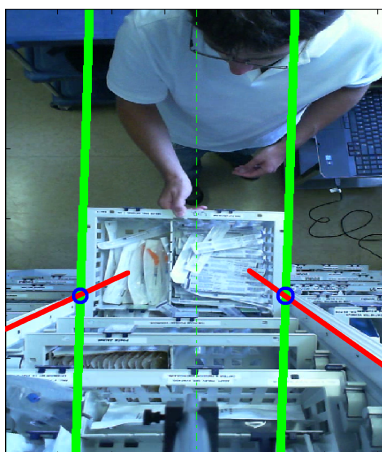


Figure 3.15 Intersections entre les côtés du tiroir et les côtés du cabinet

intersections comme dans la figure 3.15. Ensuite, lorsque vient le temps d'identifier un tiroir, il suffit de trouver la distance enregistrée qui correspond le mieux à la distance détectée. On suppose évidemment que la caméra reste fixe en tout temps et que les tiroirs ne sont pas déplacés.

3.6 Conclusion

Dans ce chapitre, nous avons vu deux approches différentes permettant de faire la détection et l'identification de tiroir. La première approche faisait l'utilisation de paires de marqueurs de couleurs. La deuxième approche se faisait sans marqueur, en utilisant des algorithmes de détection de lignes. Une méthode de correction de la distorsion a aussi été présentée afin de corriger les déformations dues par la caméra.

CHAPITRE 4

SECOND THÈME - DÉTECTION DE COMPARTIMENTS

Une fois qu'un tiroir est reconnu par notre système, on peut alors procéder à la deuxième étape principale de notre algorithme. Il s'agit de détecter les différents compartiments qui composent un tiroir. Cette étape est nécessaire avant de pouvoir estimer les quantités d'items du tiroir. En effet, l'inventaire médical est normalement organisé de façon à ce qu'un tiroir puisse contenir plusieurs types d'items différents. Pour séparer ces types d'items, on utilise des séparateurs en plastique, qui peuvent être configurés à différentes positions. Un exemple de configuration possible est montré à la figure 4.1.



Figure 4.1 Image d'un tiroir avec 5 séparateurs : 3 séparateurs horizontaux et 2 verticaux

En laissant à l'utilisateur le choix de positionner les séparateurs où bon lui semble, on lui permet de mieux optimiser l'espace de rangement disponible. Il peut arriver que certains items plus larges, comme des oreillers ou des coussins orthopédiques, utilisent tout l'espace d'un tiroir. Dans ces cas, les séparateurs ne sont simplement pas utilisés. Par contre, le fait que ces séparateurs puissent être positionnés librement ajoute un élément de difficulté à la détection d'inventaire. On ne peut alors pas supposer que ces séparateurs se trouvent à des positions fixes et il faut que le système puisse trouver leurs positions.

Dans ce projet, nous avons utilisé deux méthodes pour localiser les séparateurs d'un tiroir donné. La première méthode est de faire la détection des séparateurs par traitement d'image.

Avec cette méthode, lorsqu'un tiroir est ouvert, on suppose que les séparateurs peuvent avoir été positionnés n'importe où. Il faut donc trouver un algorithme qui puisse détecter la position de chacun des séparateurs d'un tiroir donné. Cette première méthode est décrite dans la section 4.1. La deuxième méthode consiste à indiquer préalablement au système la position de chaque séparateur de chacun des tiroirs, et à recalculer automatiquement ces positions sur les images capturées par la caméra. Cette approche requiert une intervention d'un opérateur qui saisit dans le système chacune des positions. De plus, si le personnel de l'hôpital décide de repositionner des séparateurs pour réorganiser l'inventaire, l'opérateur doit alors saisir les nouvelles positions des séparateurs. Nous verrons cette méthode dans la section 4.2.

4.1 Détection de séparateurs par transformée de Hough

4.1.1 Justification

Dans cette section, nous construisons un algorithme de traitement d'images. Le but de cet algorithme est d'identifier la position des séparateurs d'un tiroir donné. Il faut donc que l'algorithme puisse reconnaître les séparateurs à l'intérieur d'un tiroir. En traitement d'images, une approche possible pour reconnaître un objet particulier est d'utiliser ses caractéristiques physiques. Par exemple, si un objet présente une couleur particulière, il devient plus facile de l'identifier dans une scène. D'autres caractéristiques utiles peuvent être la forme de l'objet, sa texture, ou la façon dont l'objet se déplace dans l'espace comme sa vitesse ou sa trajectoire, linéaire ou non.

Dans notre cas, les séparateurs ont comme caractéristique d'être droits puisque ce sont des plaquettes en plastique. Vues de haut dans un tiroir, ces plaquettes présentent toutes une arête droite, qui reste bien visible, pourvu qu'elle ne soit pas cachée par un item. Une autre caractéristique des séparateurs est leur position. Les séparateurs ne seront jamais placés en diagonale dans un tiroir : ils seront toujours placés horizontalement ou bien verticalement, de façon à former un quadrillé.

En se basant sur ces caractéristiques physiques, on peut élaborer un algorithme de détection des séparateurs. Une étape importante de l'algorithme sera la détection de ligne, puisque c'est ce qui caractérise avant tout les séparateurs dans nos images. La revue de littérature nous permet de voir qu'il existe déjà plusieurs algorithmes de détection de ligne (Burns *et al.*, 1986; Hough, 1962; Nelson, 1994). Parmi ces algorithmes de détection de ligne, on retrouve notamment celui de la transformée de Hough. Cet algorithme est sans doute celui qui est le plus utilisé. Nous l'avons aussi utilisé dans le chapitre 3.4 pour la détection des côtés du tiroir. On peut se référer à cette section pour se rappeler comment la transformée de Hough fonctionne pour reconnaître des lignes. Aussi, il devient plus facile de réappliquer l'algorithme

ici puisqu'on s'est déjà familiarisé avec son fonctionnement et que des portions de code seront réutilisables.

4.1.2 Détection de contours

Les étapes de l'algorithme utilisé ici sont très similaires à celles de l'algorithme de détection des côtés du tiroir. Dans les deux cas, l'algorithme de Hough doit se faire à partir d'une image de contours. Dans le premier cas que nous avons étudié, celui de la détection des côtés du tiroir, nous faisons d'abord une première étape de soustraction d'arrière-plan. Après cette soustraction, nous obtenions une image binaire à partir de laquelle il était plus facile d'obtenir une image de contours. Il suffisait de prendre l'image binaire et d'utiliser un algorithme comme celui du voisinage de Moore pour obtenir les contours du tiroir.

Dans le cas présent, nous nous intéressons à la détection des séparateurs. Une soustraction d'arrière-plan ne permettrait pas de mettre en évidence les contours des séparateurs comme nous le faisons avec les côtés du tiroir. Il faut donc utiliser une autre technique pour extraire les contours des séparateurs. Une technique simple et efficace est d'appliquer un filtre de Sobel.

Le filtre de Sobel permet d'estimer le gradient de l'intensité de chaque pixel selon la direction horizontale ou verticale. Lorsque le filtre est appliqué sur un contour, on obtient alors une valeur plus élevée de gradient. Ceci s'explique parce que l'intensité varie normalement beaucoup près des contours. Pour appliquer ce filtre, on effectue une convolution sur l'image d'intensité en utilisant une des deux matrices de convolution comme dans les équations 4.1 et 4.2. Pour obtenir les contours verticaux, on utilise la première matrice de convolution alors que les contours horizontaux s'obtiennent avec la deuxième matrice. Par exemple, en utilisant l'image de départ 4.1 et en appliquant un filtre de Sobel sur son image d'intensité pour détecter les contours verticaux, nous obtenons l'image 4.2. Comme le but est de repérer les séparateurs qui se trouvent à l'intérieur du tiroir, on efface tous les contours qui se trouvent à l'extérieur du tiroir.

$$G_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * A \quad (4.1)$$

$$G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A \quad (4.2)$$

où A est l'image d'intensité sur laquelle le filtre de Sobel est appliqué.

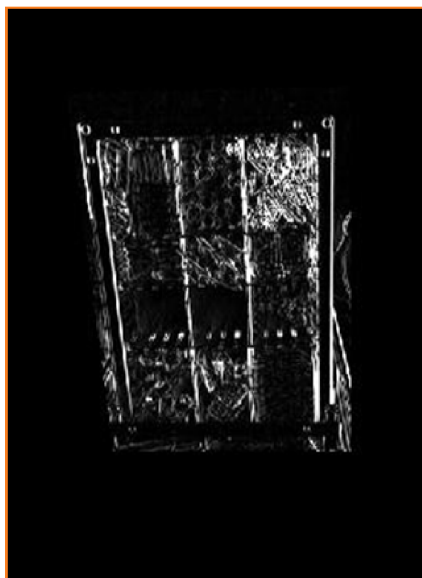


Figure 4.2 Image du tiroir avec détection de contours verticaux par une convolution de Sobel

4.1.3 Transformée de Hough en niveaux de gris

Une fois que l'image des contours horizontaux ou verticaux est obtenue, il est possible d'appliquer l'algorithme de Hough sur cette image. L'algorithme de Hough prend normalement une image binaire en entrée. Afin d'obtenir cette image binaire, il nous faudrait définir un seuil et effectuer un seuillage sur l'image en niveaux de gris. Le problème avec ce seuillage est qu'il nécessite de choisir une bonne valeur de seuil. Avec un seuil trop bas, on risque d'obtenir trop de contours et de détecter de fausses lignes. D'un autre côté, un seuil trop élevé ferait disparaître certains contours de l'image binaire et les lignes des séparateurs pourraient être mal détectées.

Pour éviter de devoir choisir ce seuil, il est possible de modifier légèrement l'algorithme de Hough. Normalement, l'algorithme est conçu pour traiter des images binaires, mais avec quelques modifications, il devient possible d'utiliser des images en niveaux de gris comme entrée. Lo et Tsai (1995) ont aussi modifié l'algorithme de Hough pour qu'il puisse prendre une image en niveaux de gris comme entrée. La démarche de l'algorithme reste la même et nous utilisons toujours un accumulateur comme à la figure 3.13 de la section 3.4. La seule différence se produit lorsque l'on doit incrémenter une case de l'accumulateur. Plutôt que d'incrémenter la case d'une seule unité, la case est incrémentée par la valeur d'intensité du pixel en question. Ainsi, un pixel avec une grande valeur d'intensité apportera des votes de plus grands poids. Un pixel avec une valeur d'intensité plus faible ne fera qu'augmenter légèrement les valeurs de l'accumulateur. Ceci fait en sorte que l'algorithme retiendra d'abord les lignes de forte

intensité plutôt que les lignes de faible intensité. Un inconvénient de cette approche est que la taille de la mémoire à réserver pour l'accumulateur doit être plus grande. Ceci est nécessaire puisqu'il faut permettre aux cases de l'accumulateur de contenir des entiers de valeurs plus élevées. Ainsi, avec cet algorithme de Hough modifié, il devient possible d'utiliser une image en niveaux de gris comme celle de la figure 4.2.

4.1.4 Choix de la taille de l'espace de Hough

L'espace de Hough est donc généré dans un accumulateur comme il a été montré à la figure 3.13. Par contre, une matrice de taille beaucoup plus grande est utilisée afin d'avoir une meilleure précision. Plutôt que d'utiliser une matrice de 11 lignes et de 9 colonnes, la taille de la matrice est augmentée selon la précision et le temps de calcul voulus.

Dans cet exemple, supposons que nous voulons une résolution de 0.5 degré sur l'angle des droites détectées. Pour détecter toutes les orientations de droites possibles, il faut avoir des valeurs d'angles allant de -90 degrés à +90 degrés. Avec ces données, on peut calculer le nombre de colonnes requises par la matrice de l'accumulateur. On effectue le calcul : $(\text{maximum} - \text{minimum}) / \text{résolution}$, c'est-à-dire $(+90 - -90)/0.5 = 360$. Ce calcul nous indique que l'accumulateur doit avoir 360 colonnes afin d'avoir une résolution de 0.5 degré.

Pour calculer le nombre de lignes requises, on doit connaître les valeurs possibles de ρ , ρ étant la distance entre l'origine et la droite. Comme l'origine se trouve dans le coin de l'image, la plus grande distance possible entre une droite détectée et l'origine est égale à la diagonale de l'image. Dans notre cas, la taille de l'image de départ est de 319 par 433 pixels. La longueur de la diagonale est donc donnée par $\sqrt{(433^2 + 319^2)} = 537.82$ pixels. Comme la valeur de ρ peut être positive ou négative selon la position de la droite, il faut prévoir que ρ pourra varier entre -537.8 et +537.8. En se fixant une résolution maximale de 1 pixel sur la valeur de ρ , on obtient donc que le nombre de valeurs possibles pour ρ est égale à $(+538 - -537)/1 = 1075$. Finalement, avec ces calculs, nous savons que nous devons allouer une matrice de 1075 lignes par 360 colonnes afin de pouvoir accumuler les valeurs de l'espace de Hough. Une fois la matrice allouée, nous pouvons accumuler toutes les valeurs requises dans la matrice selon l'algorithme de Hough en prenant l'image en niveaux de gris comme entrée. La figure 4.3 nous montre le résultat de cet algorithme. Les points foncés indiquent des valeurs élevées de la matrice alors que les points complètement blancs correspondent à des valeurs de zéro dans la matrice.

En observant la figure 4.3 obtenue par l'algorithme de Hough, on se rend compte que les valeurs maximales, c'est-à-dire les points les plus foncés, se trouvent toutes au centre de l'accumulateur. Ces points maximaux ont toujours un angle θ qui se trouve dans l'intervalle allant de -10 à +10 degrés. Ceci est normal puisque l'image de départ contient surtout des

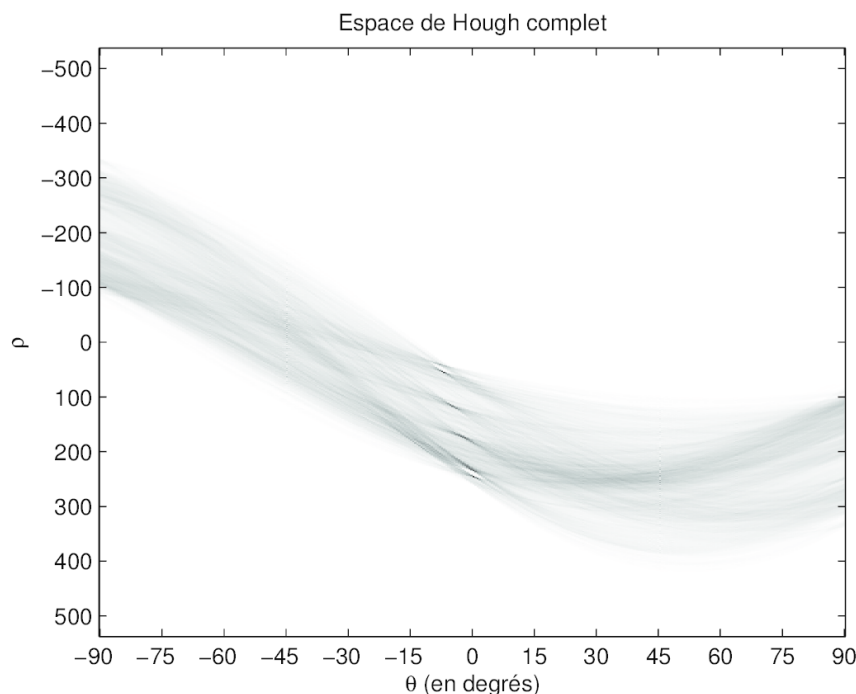


Figure 4.3 Espace de Hough de complet

lignes de séparateurs verticaux. Comme l'algorithme de Hough peut prendre un certain temps à s'exécuter selon les performances de l'ordinateur utilisé, il peut être avantageux de réduire ce temps d'exécution. Dans notre cas, on peut éviter de faire des calculs inutiles en réduisant l'espace de Hough. Plutôt que d'utiliser un accumulateur contenant toutes les valeurs d'angles possibles, on utilise un accumulateur se limitant seulement aux angles allant de -10 degrés à +10 degrés. On peut aussi réduire l'axe des ordonnées de l'accumulateur puisque les valeurs obtenues de ρ sont toujours positives. En réduisant les deux axes ρ et θ de l'accumulateur, on obtient ainsi un espace de Hough restreint. Ceci nous permet de diminuer le temps de calcul puisqu'il y a alors moins d'éléments à incrémenter dans l'accumulateur. Le résultat de ce nouvel espace de Hough est montré à la figure 4.4.

4.1.5 Identification des maxima de l'espace de Hough

Dans la figure 4.4, on peut voir plus distinctement les zones foncées. À l'œil nu, on peut compter environ 5 ou 6 sommets. Chacune de ces zones correspond à une ligne dans l'image originale. Pour faire la correspondance, des carrés de couleurs différentes sont tracés dans la figure 4.4 et les mêmes couleurs sont utilisées dans la figure 4.5. Par exemple, le carré jaune identifie un maximum qui se trouve aux coordonnées ($\theta = 0.003$, $\rho = 246$). Avec ces

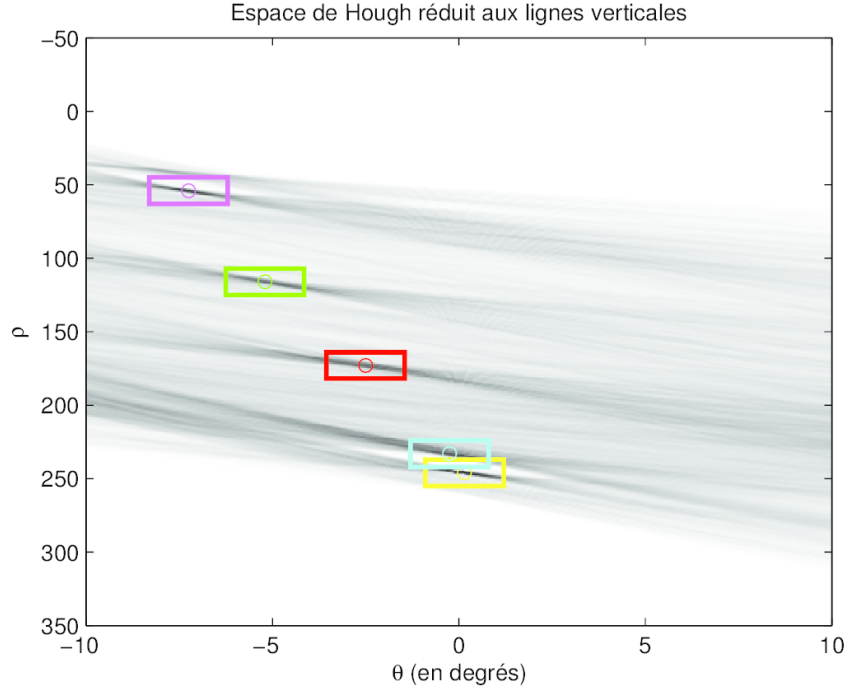


Figure 4.4 Espace de Hough réduit aux lignes verticales avec identification des maxima

valeurs de θ et ρ , on peut retrouver les coordonnées cartésiennes de la droite correspondante en utilisant l'équation 4.3 :

$$x_i \cos(\theta) + y_i \sin(\theta) = \rho \quad (4.3)$$

Comme $\cos(0.003) \cong 1$ et $\sin(0.003) \cong 0.003$, l'équation se simplifie à la relation $x_i + 0.003y_i = 246$. Le coefficient faible de 0.003 nous permet de simplifier davantage l'équation à $x_i = 246$, ce qui correspond simplement à une ligne verticale située en $x = 246$. Cette ligne est tracée en jaune dans la figure 4.5. On voit que la ligne correspond effectivement au côté droit du tiroir.

Pour trouver les autres lignes, il faut d'abord identifier les autres maxima de l'espace de Hough. Pour la première ligne, il était suffisant de chercher le maximum global de l'espace. Par contre, pour les autres lignes, il faut pouvoir trouver les maxima locaux. Une façon de procéder est d'effacer le maximum global de l'espace de Hough à chaque fin d'itération. Par exemple, à la première itération, nous identifions le maximum global marqué en jaune. Pour qu'un maximum différent soit utilisé à l'itération suivante, il suffit d'effacer ce maximum de l'espace de Hough. Pour ce faire, on prend la matrice de Hough et on assigne à zéro tous les éléments se trouvant dans le voisinage de ce maximum. Le voisinage utilisé doit avoir

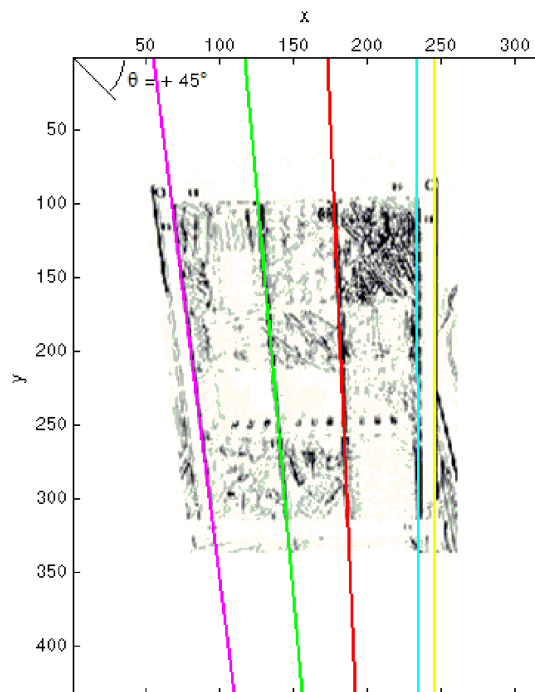


Figure 4.5 Lignes détectées dans l'image du tiroir par l'algorithme de Hough

une taille raisonnable : s'il est trop grand, on risque d'éliminer en même temps le sommet bleu de l'espace de Hough. De l'autre côté, avec un voisinage trop petit, on enlève seulement une petite portion du sommet et d'autres valeurs élevées restent présentes dans la matrice. Ceci ferait en sorte que le maximum global choisi à la prochaine itération serait situé juste à côté du maximum précédant et on détecterait deux lignes qui se trouvent très près l'une de l'autre.

La figure 4.4 montre l'identification des 5 premiers sommets par 5 itérations de notre méthode. Le nombre d'itérations est déterminé par la hauteur des sommets dans l'espace de Hough. Lorsque la valeur d'un sommet se trouve sous un certain seuil, les itérations sont terminées. Si on laissait continuer l'algorithme, les lignes trouvées ne seraient pas suffisamment grandes pour correspondre à des lignes de séparateurs. Ce seuil est choisi selon la longueur du tiroir dans l'image, puisque la longueur des séparateurs varie avec celle du tiroir.

Dans la figure 4.4, les sommets sont identifiés par un cercle. Les rectangles qui se trouvent sur chaque sommet indiquent la taille du voisinage utilisée pour effacer les éléments de la matrice. Cette taille est définie de façon arbitraire selon nos expériences. Dans notre cas, nous avons choisi une taille de voisinage pour que le premier sommet en jaune soit assez grand sans toutefois effacer le sommet bleu. Avec des tiroirs de tailles différentes ou avec des images de résolutions différentes, il faudrait redéfinir la taille de ce voisinage.

4.1.6 Détection de lignes horizontales

Maintenant que les lignes verticales sont détectées, on doit répéter la procédure afin de repérer les lignes horizontales. Le principe est essentiellement le même, mis à part le fait qu'une matrice de convolution de Sobel différente est utilisée et que l'intervalle d'angles θ définissant l'intervalle de Hough est pivoté de 90 degrés. Dans chacun des sens, on limite le nombre de lignes et le nombre d'itérations à 10, puisqu'on sait qu'un tiroir habituel ne contiendra jamais plus de 10 séparateurs horizontaux ou verticaux. Une fois que toutes ces lignes sont repérées par l'algorithme, il faut aussi faire attention à ce qu'il n'y ait pas de doublon. Si le voisinage est bien défini, ce problème ne devrait pas se produire. Les doublons peuvent toutefois être évités en inspectant les lignes obtenues et retirant celles qui croisent ou qui frôlent des lignes ayant un meilleur score dans l'espace de Hough.

Dans le cas de la détection des côtés du tiroir, on s'intéressait seulement aux deux lignes verticales du tiroir. Par contre, dans le cas ici, on veut aussi pouvoir repérer les lignes horizontales. Les lignes ne seront pas parfaitement horizontales ou verticales puisque la caméra n'est jamais parfaitement alignée. Il faut donc établir un certain seuil, en degrés, à partir duquel on considère qu'une ligne n'est plus horizontale ou verticale. Dans nos expériences, nous avons trouvé qu'un seuil de plus ou moins 20 degrés permettait d'avoir une marge suffisamment grande pour inclure toutes les lignes à détecter.

4.1.7 Délimitation des compartiments à partir des séparateurs

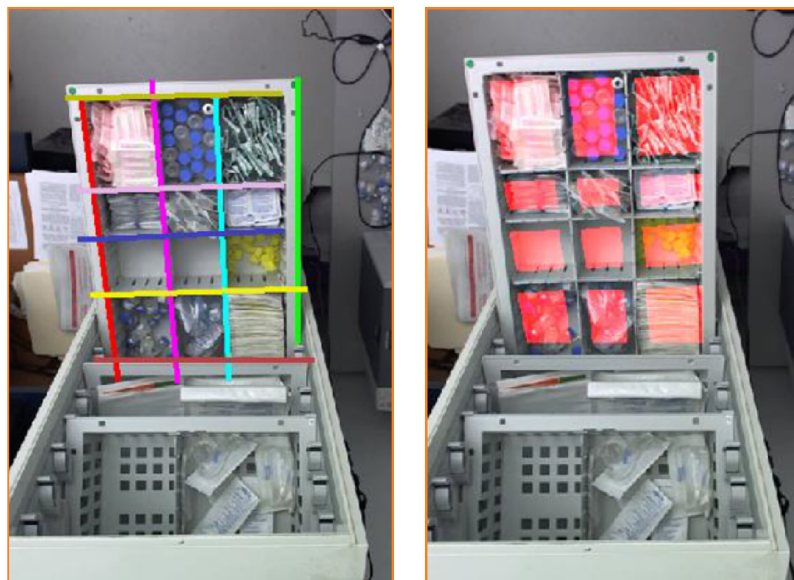
Maintenant que les séparateurs ont été reconnus par notre algorithme de détection de lignes, il reste à identifier les quatre coins de chaque compartiment. On utilise les paramètres ρ et θ de l'ensemble des lignes trouvées précédemment (figure 4.6(a)) afin d'en déduire leurs intersections. Par exemple, en connaissant les paramètres ρ et θ de deux lignes distinctes et non-parallèles, il est possible de retrouver les coordonnées x et y du point d'intersection. Pour cela, on peut utiliser les équations 4.5 et 4.6, déduites à partir de l'équation 4.4 représentant une seule droite identifiée par l'indice j . Ces équations permettent d'obtenir le point d'intersection $(x_{1,2}, y_{1,2})$ à partir de deux lignes $j = 1$ et $j = 2$.

$$x_i \cos(\theta_j) + y_i \sin(\theta_j) = \rho_j \quad (4.4)$$

$$x_{1,2} = \frac{\rho_2 \sin(\theta_1) - \rho_1 \sin(\theta_2)}{\cos(\theta_2) \sin(\theta_1) - \cos(\theta_1) \sin(\theta_2)} \quad (4.5)$$

$$y_{1,2} = \frac{\rho_2 \cos(\theta_1) - \rho_1 \cos(\theta_2)}{\sin(\theta_2) \cos(\theta_1) - \cos(\theta_2) \sin(\theta_1)} \quad (4.6)$$

Pour identifier toutes les intersections de lignes, il suffit d'effectuer ces calculs en combinant chaque ligne horizontale avec chaque ligne verticale. Avant de parcourir les lignes une à une, il est commode d'effectuer un tri selon la moyenne des coordonnées x pour les lignes verticales, ou bien selon la moyenne des coordonnées y pour les lignes horizontales. On déduit ensuite les coordonnées de chaque compartiment à partir des points d'intersections trouvées. Le résultat final est montré à la figure 4.6(b). Un rectangle rouge est inscrit à l'intérieur de chacun des compartiments.



(a) Détection des séparateurs et des côtés du tiroir (b) Délimitation des compartiments

Figure 4.6 Exemple de délimitation des compartiments à partir des séparateurs

4.2 Utilisation de la calibration pour détecter les séparateurs

4.2.1 Vue d'ensemble

Une autre approche est également possible afin de repérer la position des compartiments dans les images. Il s'agit de connaître préalablement la position physique de chaque séparateur de chacun des tiroirs. En connaissant les mesures entre chaque séparateur et le côté du tiroir, il devient possible de retrouver leurs positions dans une image. Ceci requiert que des mesures soient prises au préalable durant une phase d'installation et de calibration du système. Pour cela, on utilise une règle ou un ruban à mesurer afin d'obtenir les mesures, en centimètres, servant à décrire un tiroir. On mesure entre autres sa largeur, sa longueur et la position des séparateurs du tiroir tel qu'illustré à la figure 4.7.



Figure 4.7 Illustration de quelques mesures faites durant la calibration

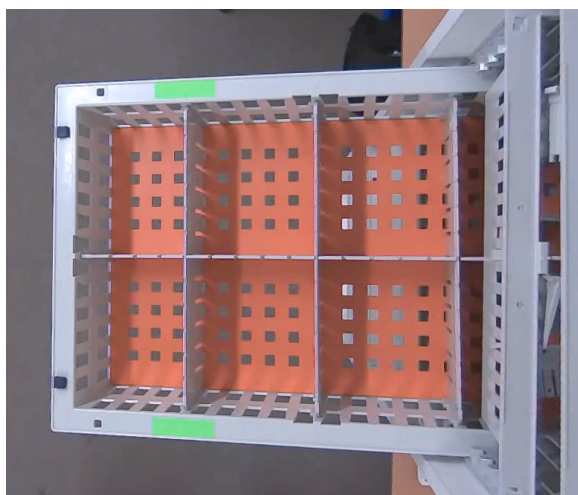
4.2.2 Détails d'implémentation

Les tiroirs que nous avons utilisés dans notre cas ont comme particularité d'avoir des séries de fentes permettant de fixer les séparateurs. La position d'un séparateur peut donc être décrite par le numéro de fente dans lequel il est placé, puisque les fentes sont toutes espacées de façon uniforme. Durant la calibration, il est effectivement plus efficace et plus précis d'identifier le numéro des fentes des séparateurs que de devoir mesurer chacune de leurs positions avec une règle ou un ruban.

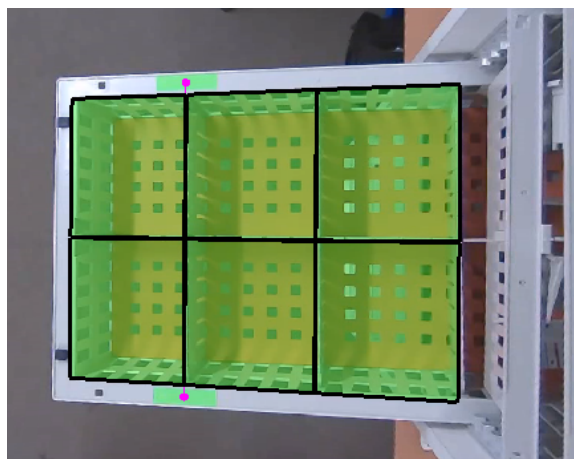
4.2.3 Procédure de recalage

Une fois la calibration faite, le système dispose de toutes les informations nécessaires pour pouvoir repérer les compartiments d'un tiroir donné. Lorsqu'un tiroir est ouvert par un utilisateur, le système s'occupe d'abord de détecter et d'identifier le tiroir par l'une des méthodes vues dans le chapitre 3. Ensuite, une fois le tiroir identifié, le système peut retrouver les distances qui décrivent ce tiroir en particulier. Pour utiliser ces distances qui sont en centimètres, le système doit d'abord convertir ces distances en pixels. Le rapport du nombre de pixels par centimètre peut être obtenu à partir de la paire de marqueurs identifiée dans l'image. Comme nous connaissons la séparation des marqueurs à la fois en centimètres et en pixels, il suffit de faire le rapport de ces deux distances pour obtenir le ratio de conversion entre centimètres et pixels. Finalement, en connaissant la position et l'orientation des marqueurs, il est possible d'utiliser les distances en pixels pour recalibrer la position de chaque séparateur du tiroir. C'est ainsi que les compartiments du tiroir peuvent être délimités.

La figure 4.8 illustre un exemple de recalage. Les marqueurs sont d'abord identifiés et deux points roses sont ajoutés à l'image. À partir de la position et de l'orientation de ces marqueurs, il est possible de délimiter les casiers du tiroir.



(a) Image de départ



(b) Casiers délimités en vert

Figure 4.8 Exemple de délimitation des casiers à partir des marqueurs

CHAPITRE 5

TROISIÈME THÈME - CLASSIFICATION DES CASIERS

Dans les chapitres 3 et 4 précédents, nous nous sommes intéressés aux problèmes de détection de tiroir et de localisation des séparateurs. Une fois ces deux premières étapes complétées, il reste une dernière étape afin de pouvoir établir l'état de l'inventaire. Cette étape est celle d'analyser les contenus des différents compartiments pour les classer comme vides ou pleins. Ces compartiments sont classifiés individuellement, en considérant chaque image de compartiment indépendamment des autres.

Pour effectuer cette classification, deux types de fonds de tiroir doivent être considérés. L'entreprise Logi-D permet à ses clients de choisir entre les tiroirs à fond lisse ou ceux à fond quadrillé. Ces deux types de fonds sont montrés dans la figure 5.1. Les fonds quadrillés ont l'avantage de moins se salir grâce aux trous carrés qui permettent de laisser passer la poussière et les débris pouvant provenir de certains items. Lorsque ces débris tombent au sol, ils sont plus faciles à ramasser pour le personnel, lors du nettoyage de la pièce.



(a) Exemple de fond lisse



(b) Exemple de fond quadrillé

Figure 5.1 Images des différents types de fond de tiroir possibles

Dans ce chapitre, nous verrons différentes méthodes de classification des casiers. Les premières méthodes que nous verrons s'appliquent surtout aux fonds de type lisse. Ces méthodes sont celle de classification par les contours à la section 5.2, celle de segmentation par croissance de région à la section 5.3, puis celle de soustraction d'arrière-plan à la section 5.4. Une méthode de détection de fonds quadrillés sera présentée à la section 5.5. Enfin, la dernière

méthode, celle de segmentation de fond coloré à la section 5.6, peut s'appliquer à la fois aux fonds quadrillés et aux fonds lisses.

5.1 Conditions préalables à la classification

La première condition préalable à la classification d'un casier est que le tiroir soit suffisamment ouvert pour montrer ce casier. Un tiroir est considéré comme ouvert lorsque ses marqueurs sont visibles et qu'il peut être identifié par une des méthodes expliquées dans le chapitre 3. Deuxièmement, l'utilisateur ne doit pas faire d'occlusion au-dessus du compartiment pour que celui-ci puisse être classifié. Tant et aussi longtemps que ces deux conditions sont respectées, le système tente de classer chacun des casiers visibles. En effet, en maximisant le nombre de trames traitées par le système, on espère améliorer les chances de pouvoir correctement estimer l'état de l'inventaire. Le traitement des trames vidéo continue jusqu'à ce que l'une de ces deux conditions ne soit plus respectée.

Durant la séquence vidéo, on s'attend normalement à ce que l'utilisateur vienne prendre ou déposer des produits médicaux dans les différents compartiments. En faisant ces manipulations, l'utilisateur fera inévitablement de l'occlusion au-dessus des compartiments, avec ses mains, ses bras, ou même sa tête s'il doit se pencher. Le problème est que pendant ce laps de temps, le système ne peut plus voir correctement le contenu des compartiments. Si le système tente de traiter un compartiment, il se pourrait qu'une main ou un bras soit confondu avec un produit de l'inventaire. Une façon pour le système d'éviter cette erreur de traitement est d'ignorer les trames où les compartiments sont masqués par l'utilisateur.

Pour savoir si une main ou un bras cache le tiroir, une solution possible serait de chercher à reconnaître les mains ou la peau de l'utilisateur. Par exemple, Jones et Rehg (2002) ont créé un modèle statistique permettant de déterminer la probabilité qu'un pixel donné soit un pixel d'une peau. De nombreux autres types de modèles existent aussi et ont fait l'objet d'une étude comparative par Vezhnevets *et al.* (2003). Cependant, tel que montré par cette étude, même les meilleurs modèles existants obtiennent toujours un taux de fausses détections d'environ 10% et un taux de bonnes détections d'au plus 94%. De plus, il peut arriver que le personnel utilise des gants médicaux ce qui fausserait alors la détection des mains.

Plutôt que de chercher à reconnaître les mains par leur couleur, il est préférable d'utiliser une autre méthode comme la détection du mouvement. Le mouvement est plus simple à détecter puisqu'il suffit de surveiller le ratio de pixels changés. Si ce ratio dépasse un certain seuil fixé à l'avance, alors on considère que du mouvement est présent à l'intérieur du tiroir observé. Ce mouvement est probablement causé par l'utilisateur puisque tous les produits de l'inventaire sont immobiles.

Si aucun mouvement n'est détecté, alors on peut supposer que les mains de l'utilisateur se trouvent à l'extérieur du tiroir, et donc que l'utilisateur ne crée aucune occlusion. C'est alors à ce moment, lorsqu'aucune occlusion n'est créée, que le système pourra tenter de classer les casiers du tiroir présentement ouvert.

5.2 Classification des casiers selon les contours d'items

5.2.1 Justification

Une première façon de faire une classification des casiers est d'utiliser les contours d'objets afin de distinguer les casiers vides des casiers non vides. On remarque en effet que le nombre de contours à l'intérieur d'un compartiment change selon le nombre et le type d'item qu'il contient. Lorsqu'un compartiment est vide, on observe alors seulement le fond du tiroir. Selon le type de fond choisi par l'hôpital, on retrouve alors un fond de type quadrillé tel que montré à la figure 5.1(b), ou bien un fond de type lisse comme dans la figure 5.1(a). En observant les casiers de cette figure, on remarque que le fond lisse est caractérisé par une absence de contours ou d'arêtes. Certains contours sont quand même présents dans un casier vide. Par exemple, dans l'image 5.2(a) la surface latérale du premier séparateur vertical devient visible à cause de l'angle de vue et de la perspective. Cette surface contient quelques encoches, ce qui fait en sorte que quelques contours sont présents dans l'image de ces deux casiers vides. De plus, il peut aussi arriver qu'un item déborde légèrement en dehors d'un casier et vienne ajouter des contours au casier adjacent. C'est ce que l'on peut observer dans le casier vide qui se trouve au centre du tiroir. Malgré les séparateurs et les items qui débordent, un casier vide avec fond lisse présentera toujours très peu de contours.

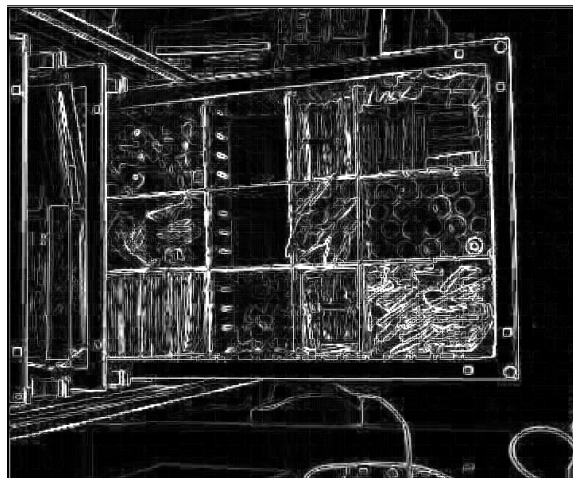
Le nombre de contours augmente lorsque des items sont ajoutés au compartiment. Selon le type d'item présent, on observe plus ou moins de contours. Les contours sont encore plus nombreux si les items sont petits ou s'ils sont enveloppés dans des sachets.

5.2.2 Explication de l'algorithme

Afin de pouvoir faire la distinction entre un compartiment vide et un compartiment plein, on utilise une méthode de détection de contours. Par exemple, il est possible d'utiliser la détection de contours de Sobel comme il a été montré dans les équations 4.1 et 4.2. Avec ces équations, on obtient le gradient selon les directions horizontale et verticale. Pour connaître la magnitude d'un pixel de contour, il suffit de calculer la norme euclidienne en utilisant les deux composantes x et y obtenues par Sobel. Pour chaque pixel, on prend le carré de la composante x ajouté au carré de la composante en y , puis on calcule la racine carrée de cette somme. Le résultat est illustré dans la figure 5.2(b). Une fois l'image de contour obtenue, on



(a) Image originale



(b) Image de la magnitude des contours obtenus par Sobel

Figure 5.2 Tiroir avec 2 casiers vides et 10 casiers pleins

effectue un seuillage afin de déterminer si un pixel donné appartient à un contour ou non. On fait cette opération de convolution et de seuillage pour chacun des casiers.

De cette façon, on peut utiliser ces images de contours pour effectuer une classification. Pour chacun des casiers, on calcule le rapport du nombre de pixels de contours sur l'aire totale en pixels du casier. Les pourcentages obtenus donnent une bonne idée du statut vide ou plein d'un casier. Par exemple, en utilisant l'image du tiroir de la figure 5.2 et un seuil de 25 sur les pixels de contour, on obtient un pourcentage d'environ 10% pour les deux casiers vides, alors que les pourcentages des autres casiers varient entre 20% et 50%. Un seuil de 15% par exemple, nous permettrait donc de faire une bonne classification des casiers vides et non vides.

5.2.3 Limitations

La méthode présentée ici comporte cependant certaines limitations. Elle suppose que les items qui se trouvent dans le tiroir présentent beaucoup de pixels de contour. Un exemple d'item bien adapté à cette méthode est celui des sachets ou bien celui de petits items avec des contours visibles comme dans la figure 5.2. Par contre, un exemple d'item qui pose problème avec cette méthode est celui de grandes boîtes de carton complètement lisses comme dans la figure 5.3. Dans ce cas, la boîte ressemble trop au fond du tiroir et le pourcentage de contours obtenu ne permet pas de faire une bonne classification.

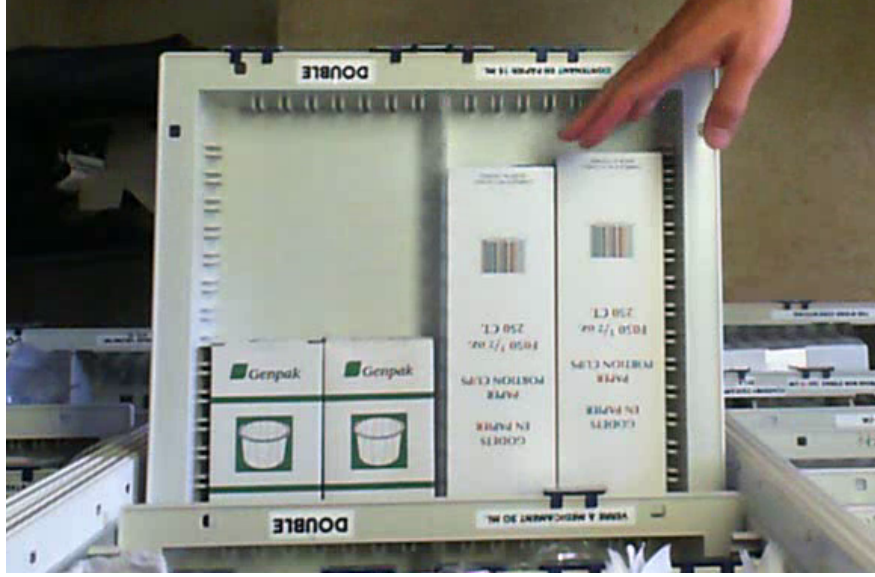


Figure 5.3 Exemple de casier contenant des boîtes d'items

5.3 Détection du contenu par la technique de croissance de région

5.3.1 Justification

Une autre façon d'effectuer la classification des casiers est d'utiliser une technique de segmentation comme celle de croissance de région ("region-growing" en anglais). Cette technique permet d'identifier une région ayant des pixels de même couleur. Dans notre cas, nous l'utilisons afin de pouvoir reconnaître les régions correspondant au fond du tiroir. La figure 5.2(a) nous montre un exemple de tiroir avec un fond lisse de la même couleur. Les deux casiers vides sont un bon exemple de fond qui pourrait être segmenté par la croissance de région.

5.3.2 Explication de l'algorithme

La première étape de cet algorithme est d'abord de choisir un point de départ dans l'image que l'on appelle graine ("seed" en anglais). On fait ensuite grandir une région à partir de cette graine tant et aussi longtemps que les pixels avoisinants ont une couleur semblable à celle de la région. On estime que deux couleurs sont semblables si la distance euclidienne entre ces deux couleurs est inférieure à un certain seuil comme dans l'équation 5.1.

$$\sqrt{(r_1 - r_2)^2 + (g_1 - g_2)^2 + (b_1 - b_2)^2} < \text{seuil} \quad (5.1)$$

La croissance de région fonctionne de façon itérative. À chaque itération, l'algorithme regarde chacun des pixels voisins pour comparer leur couleur avec celle de la région. Si un

pixel est de couleur semblable, alors on l'ajoute à la région. Ensuite, on commence une autre itération tant qu'il reste de nouveaux pixels avoisinants à considérer.



(a) Casier avec (b) Région après (c) Région après (d) Région après (e) Région une
graine de l'algo- 10 000 itérations 30 000 itérations 50 000 itérations fois l'algorithme
rithme au centre terminé
en rouge

Figure 5.4 Progression de l'algorithme de croissance de région sur l'image d'un casier après différentes itérations. La région est illustrée en vert.

La figure 5.4 illustre la progression de la croissance de région après différentes itérations. Dans la première image 5.4(a), on choisit comme point de départ le centre du casier. Dans les images suivantes, on voit la région s'agrandir avec le nombre d'itérations. Le nombre d'itérations final est élevé dans ce cas-ci puisque la résolution d'image du casier est d'environ 400 par 500 pixels, ce qui fait un total de 200 000 pixels. Finalement, dans la dernière image 5.4(e), l'algorithme se termine lorsqu'aucun des pixels voisins n'a une couleur semblable à la région. Dans cet exemple, nous avons utilisé une valeur de 10 comme seuil de l'équation 5.1.

On peut comprendre la technique de croissance de région de façon intuitive en imaginant une flaque d'eau. L'eau se propagera aux alentours seulement si la surface est assez lisse, c'est-à-dire si les couleurs voisines ne sont pas trop différentes de celle de la région.

Une fois la croissance de région terminée, la région sera plus ou moins grande selon la quantité d'items présents dans le casier. Pour pouvoir effectuer une classification d'un casier en particulier, on se base sur la taille de cette région obtenue. En divisant l'aire de la région en pixels par l'aire totale du casier, on obtient un ratio qui nous indique la proportion d'espace vide du casier. C'est avec ce ratio que l'on peut finalement déterminer si le casier est vide ou non.

5.3.3 Limitations

Cette méthode comporte toutefois certaines limitations. Comme dans la méthode précédente de classification par les contours, on suppose que les items sont suffisamment texturés. Si la texture d'un item est lisse, comme avec les boîtes de la figure 5.3, alors la région risquera

de couvrir la surface de l’item aussi. Une autre limite de la croissance de région est que le fond du tiroir ne peut pas être quadrillé comme dans la figure 5.1(b). Avec ce type de fond, la croissance de la région risque d’être bloquée par les carrés du quadrillé. Finalement, la croissance de région implique qu’une graine de départ soit choisie. Il faut faire attention à ce que ce point se trouve bien sur le fond du tiroir, sinon la région obtenue ne correspondra pas à celle recherchée.

5.4 Classification des casiers par soustraction d’arrière-plan

La classification des casiers peut aussi se faire par la méthode de soustraction d’arrière-plan. Le principe de cette méthode est de faire une soustraction entre l’image actuelle d’un compartiment et l’image d’arrière-plan du même compartiment lorsqu’il est vide. Le résultat de cette soustraction nous permet d’identifier les éléments d’avant plan qui ont été ajoutés à la scène.

5.4.1 Conditions préalables

Pour pouvoir utiliser cette méthode, il faut préalablement avoir une image de l’arrière-plan en question. Dans notre cas, l’arrière-plan de notre scène correspond à celle du compartiment vide. Ceci requiert que des images de tous les compartiments vides soient saisies lors de l’installation du système. Une fois saisie, chacune de ces images est conservée par le système afin d’être utilisée au moment où le compartiment correspondant sera ouvert par le personnel.

Certaines conditions doivent être respectées afin que cette méthode puisse fonctionner correctement. D’abord, la caméra doit toujours rester fixe afin que le point de vue de la scène reste le même. Si la caméra est déplacée, alors l’image d’arrière-plan enregistrée risquera d’être trop différente du nouvel arrière-plan réel vu par la caméra. Dans ce cas, il faudrait actualiser les images d’arrière-plan après avoir vidé complètement les compartiments. C’est pour cette raison que le support à caméra doit être suffisamment rigide afin d’empêcher tout mouvement de secousse de la caméra.

Une autre condition de cette méthode est que l’éclairage soit constant. Si l’éclairage devait changer d’intensité ou de teinte, les images d’arrière-plan ne seraient plus valides et il faudrait les actualiser, comme dans le cas précédant où la caméra avait été déplacée. Pour avoir un éclairage constant, il faut éviter de placer le système près d’une fenêtre puisque la lumière venant de l’extérieur peut changer facilement. Il faut aussi faire attention de remplacer les lumières défectueuses par le même type de lumière pour éviter de créer des changements d’éclairage. C’est seulement si les deux conditions de caméra fixe et d’éclairage constant sont respectées que la méthode de soustraction d’arrière-plan peut être utilisée.

5.4.2 Exemple de soustraction d'arrière-plan

La figure 5.5 illustre un exemple de soustraction d'arrière-plan réussi. L'image d'arrière-plan de la figure 5.5(a) est soustraite de l'image du compartiment dans la figure 5.5(b). Le résultat de cette soustraction est montré dans la figure 5.5(c). Pour segmenter l'avant-plan de l'arrière-plan, il faut ensuite faire un seuillage sur l'image obtenue. Le résultat de ce seuillage est montré dans la figure 5.5(d). Les pixels blancs indiquent des pixels d'avant plan alors que les pixels noirs sont des pixels d'arrière-plan. On peut voir que certains pixels blancs sur les séparateurs sont dus au fait que le point de vue de la caméra n'est pas exactement le même entre les deux images de départ. C'est finalement avec le ratio des pixels d'avant plan sur le nombre de pixels total d'un compartiment que nous pouvons faire une classification du compartiment.

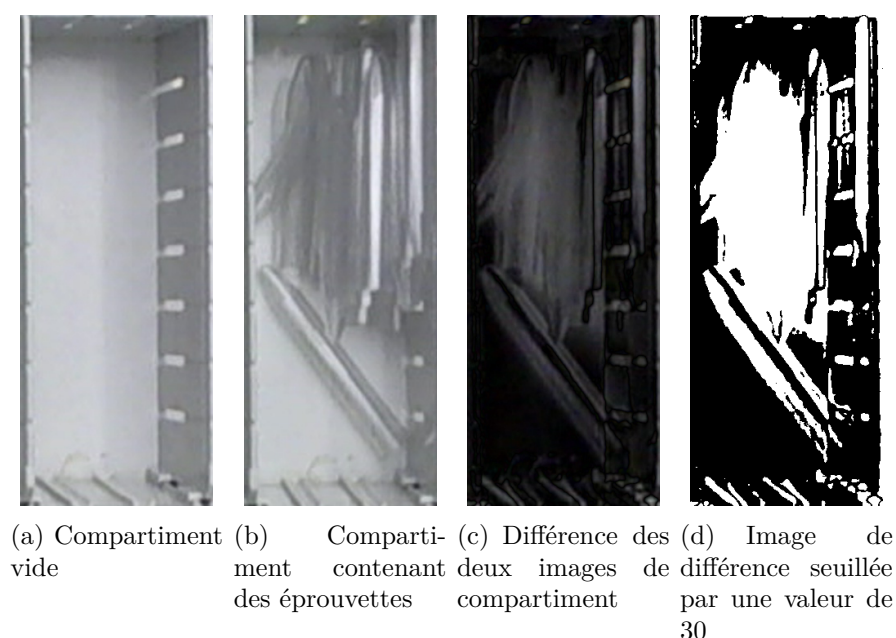


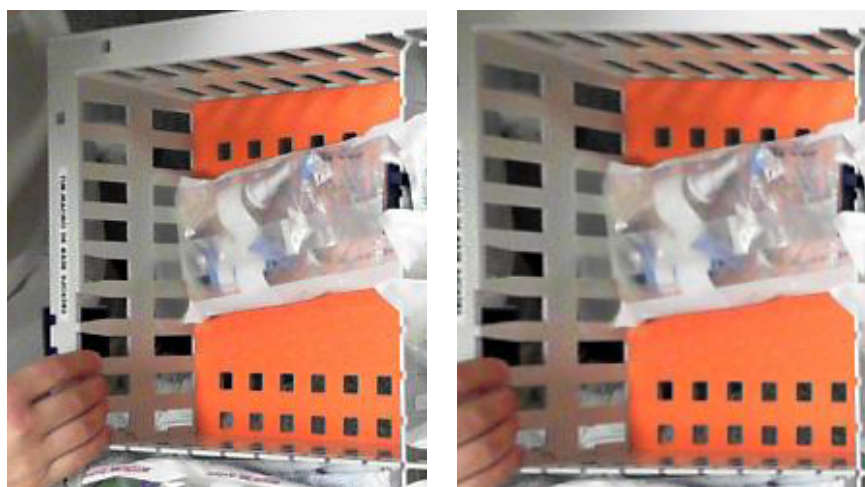
Figure 5.5 Images utilisées pour effectuer la soustraction d'arrière-plan d'un compartiment avec fond lisse

Dans la plupart des cas, l'image d'arrière-plan enregistrée reste suffisamment bien alignée avec l'image actuelle pour pouvoir effectuer une soustraction. Par contre, il peut arriver que le point de vue entre ces deux images soit trop différent pour permettre de faire une soustraction d'image précise. Par exemple, certains tiroirs sont inclinables comme dans la figure 3.7. Ce type de tiroir a l'avantage d'être plus accessible à l'utilisateur, même si le tiroir se trouve tout en haut d'un cabinet. Cependant, ce tiroir ne sera pas vu de la même façon selon son degré d'inclinaison et d'ouverture. Il se peut qu'un utilisateur assez grand se contente d'ouvrir

légèrement le tiroir sans l'incliner. Dans ce cas, le point de vue de l'image du tiroir prise par la caméra ne correspond pas à celui de l'image enregistrée où le tiroir est complètement incliné.

5.4.3 Recalage des images

Pour pouvoir effectuer une meilleure soustraction d'arrière-plan, on peut modifier notre algorithme en y ajoutant une étape préalable de recalage d'image. Le but de notre recalage est d'effectuer une déformation géométrique de l'image afin de réduire son écart avec l'image enregistrée au départ. Il s'agit en fait de faire une mise en correspondance de ces deux images afin de mieux pouvoir les comparer par la suite. La figure 5.6 montre un exemple de transformation à effectuer. L'image de gauche montre le tiroir incliné original alors que l'image de droite est corrigée par la transformation. On peut voir dans l'image corrigée que les côtés du tiroir forment un rectangle, alors que dans l'image originale, les côtés formaient plutôt un trapèze. Notre transformation d'image souhaitée peut être expliquée comme l'inverse de celui du mappage de textures en infographie tridimensionnelle (3D). En infographie, lorsqu'on crée une image de synthèse d'une scène 3D, on cherche à transformer des textures à deux dimensions (2D) pour qu'elles soient appliquées sur des objets 3D. Ici, nous voulons faire le contraire, c'est-à-dire de récupérer une texture 2D à partir d'une image 3D d'un compartiment. Au final, le but de cette transformation préalable sera de pouvoir effectuer des soustractions d'images plus justes et plus précises.



(a) Image originale d'un compartiment de tiroir incliné (b) Image transformée du compartiment

Figure 5.6 Exemple de transformation en perspective d'une image de compartiment

Il existe plusieurs techniques de recalage d'image. Certaines techniques utilisent la détection de points caractéristiques afin de mettre deux images en correspondances. Ces points

caractéristiques sont souvent des points particuliers des contours. Dans notre cas, nous pouvons utiliser comme points caractéristiques les quatre points se situant aux coins du compartiment. Ces quatre points sont connus puisque les séparateurs ont déjà été détectés, tel qu'expliqué dans le chapitre 4. Chacun de ces points est défini soit par une intersection de deux séparateurs, soit par une intersection entre un séparateur et un côté du tiroir, ou encore par le coin du tiroir. La figure 4.6(a) montre un exemple de tiroir avec la détection de ses séparateurs et de ses côtés. Une fois que toutes ces lignes sont détectées par une des méthodes présentées au chapitre 4, nous connaissons la position des quatre coins pour chacun des compartiments du tiroir.

Pour effectuer cette transformation, il est nécessaire d'utiliser quelques outils mathématiques provenant de l'algèbre linéaire. D'abord, le type de transformation linéaire en question est en fait une transformation dite matricielle, parce qu'elle est représentable par une équation matricielle. Dans notre cas, la matrice de transformation T utilisée est de taille 3×3 et contient donc 9 valeurs inconnues a données dans l'équation 5.2. C'est cette matrice qui permet définir la relation géométrique qui existe entre les points de l'image originale d'entrée et les points de l'image de sortie.

$$T = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad (5.2)$$

On utilise cette matrice en la multipliant par les coordonnées u et v d'un point de l'image originale comme dans l'équation 5.3. Le résultat de cette multiplication nous permet d'obtenir les coordonnées homogènes x' , y' et w' du même point dans l'image corrigée. Comme ce sont des coordonnées homogènes, on doit diviser chacune des composantes x' et y' par la dernière composante w' (équations 5.4), afin d'obtenir les vraies coordonnées x et y du point. Wolberg (1990) décrit les différents types de transformations affines possibles grâce à cette matrice T . Ces différentes transformations peuvent être des translations, des rotations, des mises à échelle, des cisaillements ou des projections. Dans notre cas, nous nous intéressons à la transformation la plus générale, c'est-à-dire la projection.

$$\begin{bmatrix} x' & y' & w' \end{bmatrix} = \begin{bmatrix} u & v & 1 \end{bmatrix} * \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad (5.3)$$

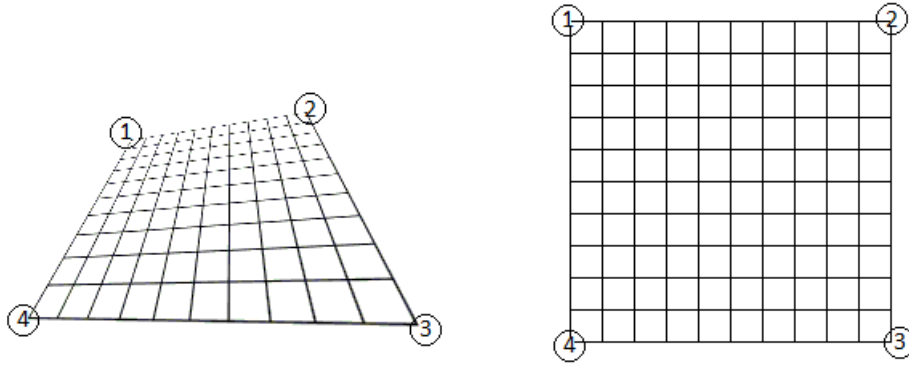
$$x = x'/w' \text{ et } y = y'/w' \quad (5.4)$$

La transformation géométrique étant définie seulement par une matrice 3×3 , il suffit de trouver les valeurs des 9 inconnues de cette matrice. On peut réduire le nombre d'inconnues en normalisant la matrice T , c'est-à-dire en divisant tous ses éléments par a_{33} . L'élément a_{33} est uniquement responsable de la mise à l'échelle, donc on pourrait même l'ignorer dans notre cas. En faisant cette normalisation, l'élément a_{33} prend la valeur de 1, et ceci nous laisse maintenant avec 8 degrés de liberté.

À présent, comme il y a 8 inconnues dans l'équation matricielle, il faudra 8 équations afin de trouver ces 8 inconnues. Chaque correspondance de points (x, y) et (u, v) permet de créer un ensemble de 2 équations puisqu'il y a une équation par composante. Les équations 5.5 et 5.6 nous montrent la forme que prennent ces équations pour chacune des deux composantes. Elles sont obtenues à partir de l'équation matricielle 5.3. Comme chaque point génère 2 équations et il faudra donc 4 points afin d'avoir 8 équations. Ces 4 points doivent être non colinéaires afin que la résolution du système d'équations donne une solution unique. La figure 5.7 illustre un exemple de choix de 4 points pour une image d'un quadrillé. Le quadrillé permet de mieux comprendre l'effet de cette transformation de perspective.

$$x = \frac{x'}{w'} = \frac{a_{11}u + a_{21}v + a_{31}}{a_{13}u + a_{23}v + a_{33}} \quad (5.5)$$

$$y = \frac{y'}{w'} = \frac{a_{12}u + a_{22}v + a_{32}}{a_{13}u + a_{23}v + a_{33}} \quad (5.6)$$



(a) Image d'un quadrillé déformée par la perspective
(b) Image du même quadrillé corrigée par une transformation en perspective

Figure 5.7 Exemple de transformation en perspective d'une image quadrillée. Les 4 coins numérotés servent à trouver la matrice de transformation.

On peut réorganiser ces équations afin de faciliter la résolution du système d'équations. Dans les équations 5.5 et 5.6, on remplace d'abord a_{33} par 1 grâce à la normalisation de la matrice T . Ensuite, on élimine la fraction en multipliant chacun des deux membres de l'équation par le dénominateur : $a_{13}u + a_{23}v + 1$. Ceci nous permet d'obtenir les équations 5.7 et 5.8 qui ne contiennent aucune fraction ou division. Puisque nous avons 4 points d'entrée (x_k, y_k) et 4 points transformés (u_k, v_k) , on identifie chacun de ces points par l'indice k ($k = 0, 1, 2, 3$). Les premiers points s'écrivent donc (x_0, y_0) et (u_0, v_0) . En répétant les équations 5.7 et 5.8 4 fois pour chacun des 4 points, on obtient un système de 8 équations qui peut aussi être réécrit sous la forme d'une seule équation matricielle. Cette nouvelle équation 5.9 prend la forme $Ax = B$ où x représente l'ensemble des inconnues a de la matrice T .

$$x = a_{11}u + a_{21}v + a_{31} - a_{13}ux - a_{23}vx \quad (5.7)$$

$$y = a_{12}u + a_{22}v + a_{32} - a_{13}uy - a_{23}vy \quad (5.8)$$

$$\begin{bmatrix} u_0 & v_0 & 1 & 0 & 0 & 0 & -u_0x_0 & -v_0x_0 \\ u_1 & v_1 & 1 & 0 & 0 & 0 & -u_1x_1 & -v_1x_1 \\ u_2 & v_2 & 1 & 0 & 0 & 0 & -u_2x_2 & -v_2x_2 \\ u_3 & v_3 & 1 & 0 & 0 & 0 & -u_3x_3 & -v_3x_3 \\ 0 & 0 & 0 & u_0 & v_0 & 1 & -u_0y_0 & -v_0y_0 \\ 0 & 0 & 0 & u_1 & v_1 & 1 & -u_1y_1 & -v_1y_1 \\ 0 & 0 & 0 & u_2 & v_2 & 1 & -u_2y_2 & -v_2y_2 \\ 0 & 0 & 0 & u_3 & v_3 & 1 & -u_3y_3 & -v_3y_3 \end{bmatrix} * \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \\ a_{12} \\ a_{22} \\ a_{32} \\ a_{13} \\ a_{23} \end{bmatrix} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} \quad (5.9)$$

Si l'on veut trouver les inconnues a de la matrice T , il faut donc résoudre cette équation matricielle de la forme $Ax = B$. Comme avec toute autre équation de ce type, on résout l'équation en isolant la matrice d'inconnues. Pour ce faire, on calcule d'abord l'inverse de la matrice A . Une fois la matrice inverse trouvée, nous modifions l'équation de façon à isoler x et nous obtenons $x = A^{-1}B$.

On obtient donc les éléments a de la matrice de transformation T en résolvant l'équation 5.9 selon la formule $x = A^{-1}B$. On réarrange ensuite les éléments a obtenus dans une matrice 3×3 afin d'obtenir la matrice de transformation T 5.2. Il existe aussi une méthode de résolution plus rapide que celle d'inverser la matrice A . Cette méthode suppose cependant que le résultat de la transformation du compartiment soit un carré, ou un rectangle. Dans ce cas, certaines simplifications algébriques sont possibles ce qui permet d'éviter de devoir inverser une matrice. Les équations de cette méthode sont expliquées par Wolberg (1990).

En multipliant cette matrice T par chaque point d'une image comme dans l'équation 5.3, on peut transformer une image d'un compartiment et ainsi réduire l'effet de perspective. Un exemple de transformation est montré à la figure 5.6. Comme on redimensionne en quelque sorte l'image du compartiment, il faut utiliser une méthode d'échantillonnage afin de produire l'image finale. Des méthodes possibles sont l'interpolation linéaire, bilinéaire ou cubique. Dans notre cas, nous avons choisi la méthode d'interpolation bilinéaire.

L'étape préalable de transformation d'image est maintenant complétée et nous avons une image corrigée. C'est avec cette image corrigée que nous pouvons appliquer l'algorithme de soustraction d'arrière-plan que nous avons décrit et illustré à la figure 5.5. Les mêmes conditions d'éclairage constant et de caméra fixe doivent toujours être respectées afin que l'algorithme fonctionne correctement.

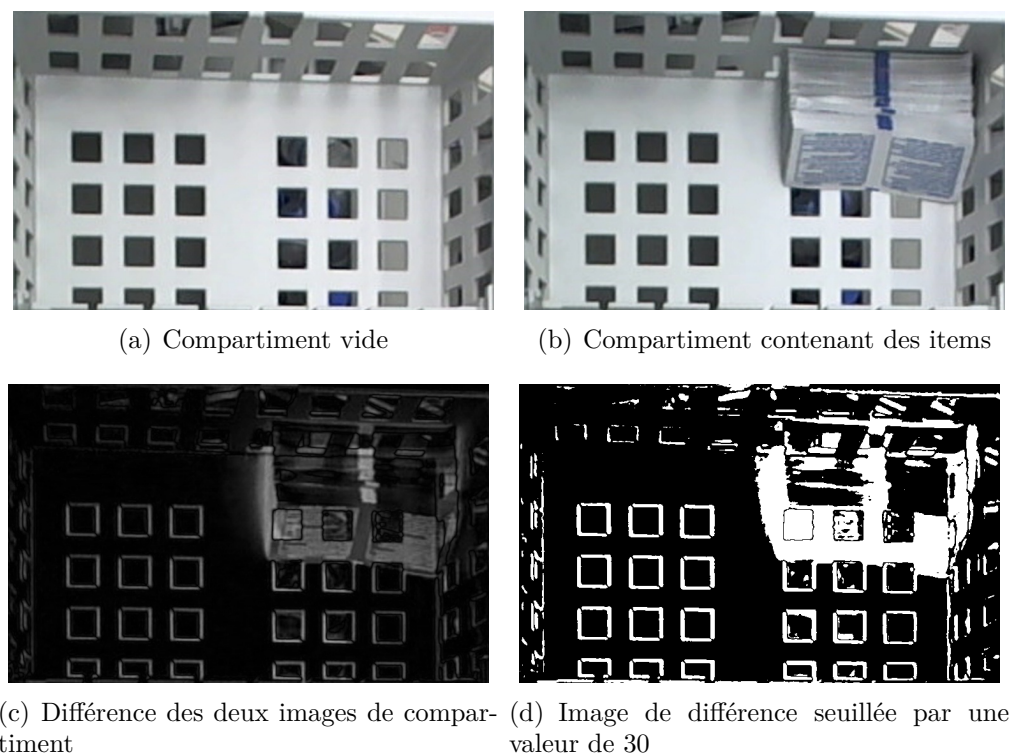


Figure 5.8 Images utilisées pour effectuer la soustraction d'arrière-plan d'un compartiment avec fond quadrillé

5.4.4 Limitations

La méthode de soustraction d'arrière-plan décrite dans cette section comporte toutefois certaines limitations. D'abord, elle requiert qu'une grande quantité d'images d'arrière-plan soit enregistrée lors de la calibration du système. Cette procédure peut être longue à com-

pléter, surtout pour un hôpital avec beaucoup de cabinets. Une deuxième limitation est que cette méthode fonctionne surtout avec des fonds lisses. Lorsque les fonds sont quadrillés comme dans la figure 5.8, l'algorithme détecte alors les quadrillés comme étant des items d'inventaire. Ceci est dû au fait que l'image d'arrière-plan n'est jamais parfaitement alignée avec l'image actuelle, malgré l'étape de transformation d'image. Cette légère différence d'alignement suffit pour causer l'apparition de carrés dans l'image de différence comme celle de la figure 5.8(d). Ces deux principales limitations à la méthode de soustraction d'arrière-plan sont suffisantes pour nous motiver à chercher une meilleure méthode de classification de casiers.

5.5 Classification des casiers par détection de fonds quadrillés

5.5.1 Justification

Les méthodes de classification que nous avons vues jusqu'à présent s'appliquaient surtout aux fonds de tiroir lisse. Il y a aussi un autre type de fond à considérer, celui des fonds quadrillés. Ce type de fond est aussi utilisé par les hôpitaux parce qu'il permet de réduire la fréquence de nettoyage des tiroirs. Grâce aux quadrillés, la poussière venant des items peut passer par les trous et tomber au sol. La figure 5.9 montre un exemple de ce fond pour un seul casier.



Figure 5.9 Exemple de casier avec un fond de type quadrillé

Notre algorithme doit aussi être capable d'effectuer une classification sur ce type de fond quadrillé. Les autres méthodes vues jusqu'ici ne fonctionneraient pas bien dans ce cas à cause

de la présence des carrés. On peut toutefois utiliser ces carrés à notre avantage pour obtenir un nouvel algorithme de classification. Si un algorithme réussit à détecter les carrés du fond du tiroir, il sera alors possible de faire une classification à partir de cette détection.

5.5.2 Détection de carrés par transformée de Hough généralisée

La première étape de l'algorithme décrit dans cette section devra donc être de détecter les carrés du fond du tiroir. Il existe déjà plusieurs méthodes dans la littérature pour détecter des carrés ou des rectangles. Par exemple, une méthode connue est celle de la transformée de Hough généralisée. Comme la transformée de Hough que nous avons utilisée précédemment pour détecter des lignes, l'algorithme de transformée de Hough généralisée fonctionne de façon similaire. Elles utilisent toutes les deux un accumulateur servant à comptabiliser les votes pour chacune des dispositions possibles de la forme recherchée dans l'image. Dans le cas où nous recherchions des lignes dans l'image, l'accumulateur utilisé n'avait que deux dimensions. La première dimension ρ donnait la distance entre la ligne et l'origine du système de coordonnées alors que la deuxième dimension θ donnait l'orientation de la ligne. Dans la situation où nous voudrions détecter des formes plus complexes que des lignes, comme des carrés ou des cercles, il faudrait alors ajouter des dimensions supplémentaires à l'accumulateur. Par exemple, un carré parfait peut être tracé dans une image selon au moins 4 degrés de liberté. Deux premiers degrés de liberté correspondent aux composantes x et y du centre du carré. Deux autres degrés de liberté pourraient être la largeur des côtés du carré, puis son orientation, puisque le carré peut pivoter sur lui-même.

L'accumulateur de la transformée de Hough devra donc avoir 4 dimensions, c'est-à-dire 2 de plus que dans l'algorithme original de Hough. Le nombre de dimensions de l'accumulateur est un facteur important à considérer pour des raisons de performance. En effet, chaque dimension ajoutée augmente la mémoire et le temps de calcul requis par l'algorithme de façon exponentielle. L'option d'utiliser la méthode généralisée de Hough devient alors moins attrayante pour notre application, parce que le traitement des images doit être complété dans un délai raisonnable. C'est pour cette raison que nous avons décidé d'explorer d'autres méthodes que celle-ci pour détecter les quadrillés du tiroir.

5.5.3 Autres méthodes de détection de carrés

La littérature nous offre de nombreux articles traitant de la détection de carrés ou de rectangles. La détection de ces formes est un problème courant avec plusieurs applications possibles. Par exemple, elle peut servir à la détection de bâtiments dans des images satellites ou aériennes (Tao *et al.*, 2002). Un autre exemple est celui de la détection de plaques

d'immatriculation d'un véhicule (Gao et Zhou, 2000). Selon les besoins et les contraintes des différentes applications, les auteurs ont élaboré plusieurs approches pour détecter des formes rectangulaires.

Par exemple, Ioannou et Dugan (1996) se sont intéressés au problème un peu plus général de détection de parallélogrammes. Leur méthode utilise d'abord la transformée de Hough pour détecter chacune des lignes pouvant faire partie d'un parallélogramme. Ensuite, leur deuxième étape est un algorithme qui estime la longueur des lignes trouvées, puisque l'algorithme de Hough ne donne pas cette information. Une fois que les lignes ainsi que leurs longueurs sont obtenues, leur méthode tente de combiner ces lignes selon certains critères comme la distance entre les lignes et les angles formés. Les résultats de cette méthode sont meilleurs lorsque des images synthétiques sont utilisées. Dans ces images, on peut facilement contrôler le niveau de bruit, c'est-à-dire le nombre d'objets autres que des parallélogrammes. Une image réelle est aussi utilisée, mais elle contient uniquement deux boîtes posées contre un mur. Ce genre de scène simple ne correspond pas très bien à notre situation où nous pourrions avoir beaucoup d'items médicaux, causant du bruit autour des carrés à détecter.

Jung et Schramm (2004) utilisent des images qui ressemblent un peu plus à notre cas, puisque des objets non rectangulaires sont ajoutés aux scènes. La transformée de Hough est également utilisée pour détecter les lignes de façon individuelle. Par contre, plutôt que de calculer la transformée de Hough une seule fois sur l'image complète, la transformée est appliquée plusieurs fois, sur une seule portion de l'image à la fois. Une fenêtre de recherche glissante est définie et est utilisée pour balayer l'ensemble de l'image de façon exhaustive. À chaque déplacement de cette fenêtre, on recalcule alors la transformée de Hough. La taille de cette fenêtre est choisie en fonction de la taille des objets rectangulaires recherchés. La fenêtre doit être suffisamment grande pour pouvoir contenir complètement l'objet le plus grand à détecter. L'avantage de cette approche est qu'elle permet de réduire le bruit causé par les objets avoisinants. Une fois que Jung et Schramm (2004) ont identifié des lignes dans l'espace de Hough, ils tentent d'apparier ces lignes entre elles directement depuis l'espace de Hough selon certaines contraintes géométriques.

Nguyen *et al.* (2009) utilisent aussi une fenêtre de recherche pour faire le suivi d'un objet dans une séquence vidéo. Par contre, comme le but de l'application est de détecter un seul objet rectangulaire, la fenêtre est utilisée différemment. Plutôt que de balayer complètement chacune des trames de la vidéo, la fenêtre est positionnée en fonction des détections effectuées dans les trames précédentes. Une position de départ est probablement définie pour la première trame de la vidéo. En plus d'éliminer le bruit se trouvant à l'extérieur de la fenêtre de recherche, Nguyen *et al.* (2009) proposent d'effacer le bruit qui peut se trouver à l'intérieur du rectangle recherché. Pour ce faire, un cercle est défini au centre de la fenêtre de recherche

et tous les pixels qui se trouvent à l'intérieur de ce cercle sont effacés avant d'appliquer la transformée de Hough. Une fois les lignes trouvées, on tente encore ici de les apparier selon des contraintes géométriques comme le parallélisme.

L'application développée par Tao *et al.* (2002) est probablement celle qui ressemble le plus à la nôtre. On peut comparer la détection de bâtiments rectangulaires dans les images aériennes avec la détection des carrés dans nos images de tiroirs. Dans les deux cas, les objets rectangulaires à détecter peuvent être en basse résolution, parce que la résolution offerte par la caméra ne suffit pas toujours. Tao *et al.* (2002) ont préféré utiliser un filtre de Canny suivi de certains calculs d'arithmétiques pour détecter les lignes. Ceci permet sûrement de réduire la quantité de calculs puisqu'aucune fenêtre glissante ne semble être utilisée. Ensuite, le reste de l'article porte surtout sur un algorithme de combinaison entre différents types de lignes et de coins.

D'après les articles que nous avons résumés ici, nous pouvons en dégager quelques étapes qui sont communes aux différentes approches de détection de rectangles :

1. Binariser l'image de départ par une méthode de détection de contours, que ce soit avec un filtre de Canny, un filtre de Sobel, ou un autre filtre.
2. À partir de l'image de contours, extraire les lignes par une méthode de détection comme la transformée de Hough.
3. Une fois les lignes obtenues, chercher les combinaisons de lignes qui respectent certains critères géométriques comme le parallélisme ou la distance entre elles.

Notre but sera d'appliquer cette méthode au problème de détection des carrés du tiroir.

5.5.4 Extraction de contours

Pour illustrer le traitement fait par notre algorithme, une image de départ doit d'abord être choisie. Prenons par exemple l'image de la figure 5.10(a), qui est en fait une sous-image de la figure 5.9.

Pour accomplir la première étape de notre méthode, celle de binariser l'image, on commence d'abord par calculer l'image des gradients comme à la figure 5.10(b). Cette image des gradients nous indique, pour chaque point de l'image, la direction de la plus grande variation d'intensité. La grandeur de chaque vecteur bleu dans la figure indique la vitesse de la variation d'intensité à ce point. Avec cette image, on remarque que les plus grandes variations d'intensité se trouvent souvent sur les contours des objets. Pour en extraire une image de contours, il suffit donc de prendre l'image des gradients et de seuiller chaque gradient selon la grandeur de sa norme. Avec ce seuillage, nous pouvons obtenir l'image binaire de la figure 5.10(c). Ceci vient compléter la première étape de notre méthode.

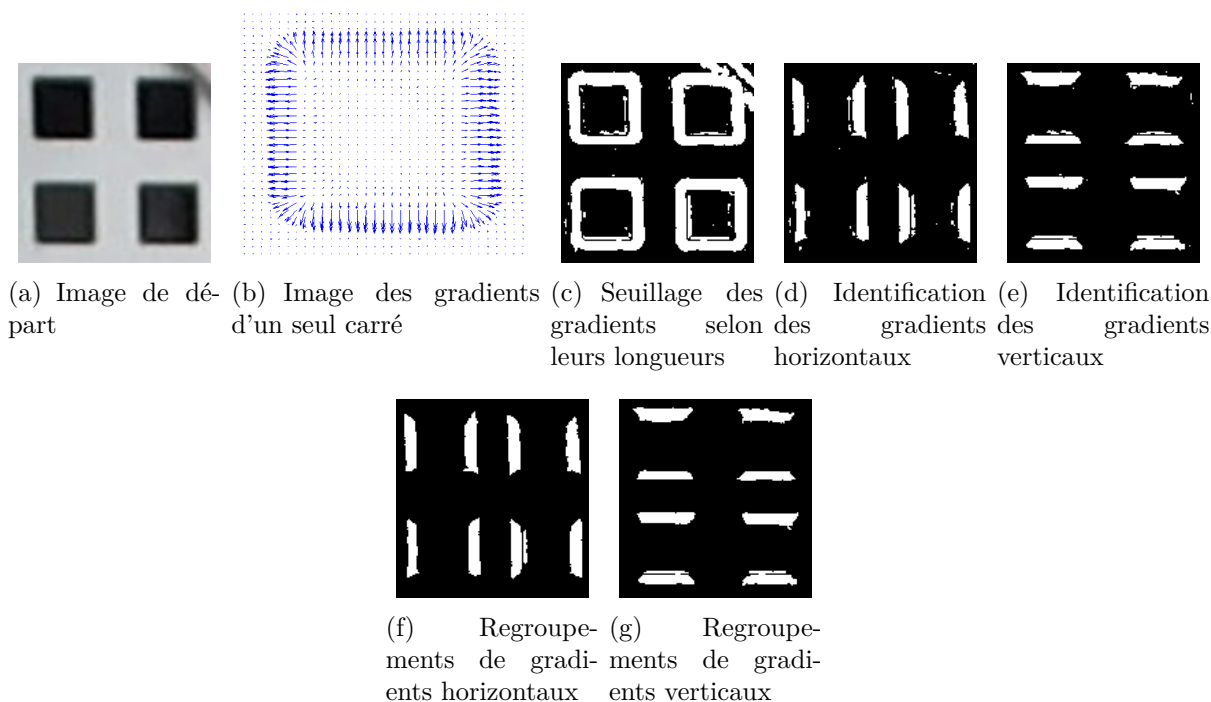


Figure 5.10 Étapes de détection des lignes verticales et horizontales

5.5.5 Détection des segments de carrés par l'algorithme de Burns

À partir de l'image de contours obtenue, il devient plus facile d'appliquer un algorithme de détection de lignes. Un choix de quelques algorithmes s'offre à nous. Il y a d'abord l'algorithme de Hough qui est bien connu et que nous avons utilisé pour d'autres aspects de ce projet. Cet algorithme se prête bien à la détection des séparateurs ou des côtés des tiroirs, parce que ces lignes sont relativement longues par rapport au reste de l'image. La longueur de ces lignes fait en sorte que le bruit causé par les autres objets a très peu d'effet sur la réussite de la détection. Dans le cas présent où nous voulons détecter des lignes plus petites, leur détection risque d'être facilement influencée par le bruit environnant. Pour éviter ce problème, Jung et Schramm (2004) proposent de réduire l'effet du bruit en utilisant une fenêtre de recherche. Cette fenêtre doit être environ de la même taille que l'objet à rechercher pour être efficace. Par contre, l'inconvénient avec l'utilisation d'une fenêtre est que cela duplique la quantité de calculs à effectuer ; puisque pour chaque position de fenêtre, il faut alors recalculer l'accumulateur de Hough. Cet algorithme, avec la fenêtre glissante, permettrait probablement de détecter les lignes correctement, mais son exécution serait plus longue. Il serait préférable d'avoir un algorithme qui puisse effectuer la détection plus rapidement.

La littérature scientifique nous présente d'autres méthodes que celle de Hough pour la détection de lignes. Une méthode intéressante est celle introduite par Burns *et al.* (1986).

Cette méthode s'applique surtout à la détection de segments de lignes ou de lignes plutôt courtes. C'est justement ce type de ligne que nous voudrions détecter dans ce cas-ci.

Pour identifier des segments, l'algorithme de Burns se base sur l'information des gradients de chaque pixel. Le calcul des gradients doit d'abord être fait comme à la figure 5.10(b). Ensuite, seulement les gradients suffisamment grands sont considérés et les gradients plus petits ne sont pas utilisés. Ce seuillage des gradients est illustré à la figure 5.10(c). L'idée principale de l'algorithme de Burns est de classer chacun des gradients en partitions selon leur orientation. Pour chacun des gradients, on calcule son orientation puis on l'assigne à l'une des 8 partitions de la figure 5.11. Par exemple, les gradients qui se trouvent sur des segments de lignes verticales seront classés dans les partitions 1 et 5. Les gradients de segments horizontaux se trouveront quant à eux dans les partitions 3 et 7. Le résultat de ces classifications est montré dans les figures 5.10(d) et 5.10(e). Supposons que l'image présentait des segments de ligne diagonaux et que nous voudrions les détecter, on utiliserait alors les partitions 2, 4, 6 et 8.

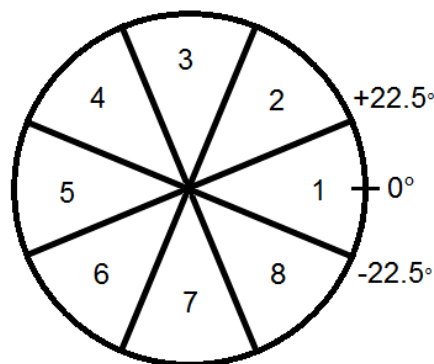


Figure 5.11 Partitionnement des gradients selon leur orientation par l'algorithme de Burns

La dernière étape de l'algorithme de Burns est de regrouper les pixels adjacents qui appartiennent à une même partition. Ces regroupements de pixels (ou “blobs” en anglais) prennent normalement une forme allongée correspondant aux segments de ligne détectés. Certains de ces regroupements ne correspondent pas toujours à des lignes, mais sont plutôt causés par du bruit. Par exemple, dans les figures 5.10(d) et 5.10(e), on peut remarquer la présence de quelques pixels blancs isolés. Pour que ces pixels ne soient pas considérés comme des lignes par l'algorithme, on peut faire en sorte que seulement les regroupements suffisamment grands soient sélectionnés. Si un regroupement ne contient pas un certain nombre de pixels minimal fixé au départ, alors il est enlevé de la liste des regroupements à considérer. Dans notre exemple, si l'on applique ce critère de sélection de taille minimale, puis que l'on affiche seulement les regroupements de pixels conservés, on obtient alors les figures 5.10(f) et 5.10(g).

On peut voir dans ces deux figures que les pixels isolés ont disparus et que seulement les regroupements correspondant à des lignes sont restés.

Les figures 5.10(f) et 5.10(g) nous montrent un exemple de détection de segments bien réussie par l'algorithme de Burns. Cependant, cet exemple reste assez simple puisque l'image de départ ne présente que des carrés et aucun autre objet. En pratique, les images contiendront d'autres objets que des carrés et le nombre de segments sera plus nombreux. Pour rendre la détection de Burns plus robuste, on pourrait ajouter d'autres critères de sélections sur les segments. Par exemple, un critère possible pourrait être celui d'une longueur minimale ou maximale de segment. On peut estimer à l'avance quelle taille auront les segments de carrés, si l'on connaît la taille du tiroir en question. En effet, le rapport entre la taille des segments et la largeur du tiroir sera toujours le même, peu importe la hauteur du tiroir utilisée. La figure 5.12 nous montre les lignes en rouge qui ont été détectées par l'algorithme de Burns et qui respectent les longueurs minimales et maximales.



Figure 5.12 Détection des segments de lignes par Burns et identification des carrés par des points verts

Un avantage de l'algorithme de Burns par rapport à celui de Hough est qu'il permet d'obtenir la longueur des segments de ligne plus facilement. Une fois le regroupement obtenu, il suffit de calculer sa boîte englobante. L'algorithme de Hough nous permettait seulement d'obtenir l'orientation et la position d'une ligne. Pour obtenir la longueur de la ligne, il fallait reprendre l'image des contours et regarder quels pixels de contour se trouvaient sous la ligne en question.

5.5.6 Combinaisons des segments détectés en carrés

Maintenant que les segments de lignes ont été extraits de l'image, il ne reste plus que la troisième étape de notre méthode. Cette étape vise à chercher les combinaisons de segments qui pourraient satisfaire les critères d'un carré. Pour ce faire, on cherche d'abord à combiner les segments de façon à former des coins de carré. Pour trouver ces coins, on parcourt exhaustivement chacune des paires possibles de lignes, faites d'une ligne verticale et d'une ligne horizontale. On considère alors qu'un coin de carré est formé si deux critères sont respectés. Le premier critère est que ces lignes doivent être de longueur semblable, puisque les côtés d'un carré sont toujours égaux. Pour être de longueur semblable, le rapport de leurs longueurs doit se trouver suffisamment près de 1, c'est-à-dire que la différence entre 1 et ce rapport doit être plus petite qu'un certain seuil donné. Le deuxième critère est que ces lignes doivent se rejoindre en un point. On considère que deux lignes se rejoignent en un point si la distance entre les extrémités de chacune de ces lignes est plus petite qu'un certain seuil. Si ces deux critères sont respectés, nous considérons alors que la paire de segments forme un coin.

Par contre, un seul coin n'est pas suffisant pour former un carré. Ce coin doit au moins être combiné à un autre coin pour qu'il puisse être considéré comme un carré. La dernière étape est donc de combiner des coins entre eux pour former des carrés. Le seul critère pour pouvoir combiner deux coins est qu'ils doivent partager une même ligne. Dès que deux coins sont combinés, cela signifie en fait que nous avons déjà les trois côtés d'un carré, puisqu'un de ces côtés est partagé par les deux coins. Le quatrième côté n'a pas vraiment besoin d'être cherché, puisque l'on peut déduire la position d'un carré avec seulement trois côtés. La figure 5.12 nous montre les carrés qui ont été identifiés par cette méthode. On voit dans cet exemple que les 6 carrés présents ont été détectés avec succès.

5.5.7 Croissance de région à partir des carrés détectés

Le nombre de carrés détectés nous permet d'estimer la portion vide d'un casier. En connaissant la taille d'un carré, on peut avoir une idée de l'aire occupée par l'ensemble des carrés détectés. Pour connaître le pourcentage du casier qui est vide, on pourrait donc diviser l'aire occupée des carrés par l'aire totale du casier. Ce calcul fonctionnerait bien dans l'exemple précédant. Par contre, ce calcul se prêterait mal à certains casiers comme celui de la figure 5.13. En effet, il arrive parfois qu'une partie du fond ne soit pas quadrillée, mais lisse comme dans la portion droite de ce casier. Dans ces cas-ci, la seule détection des carrés ne nous permettrait pas d'estimer correctement la proportion vide du casier. Si l'on tente d'effectuer la détection des carrés et que l'on effectue le calcul de division d'aires par la suite,

on obtiendra alors comme résultat que la partie lisse du casier est toujours occupée, parce qu'aucun carré n'est présent dans cette région.

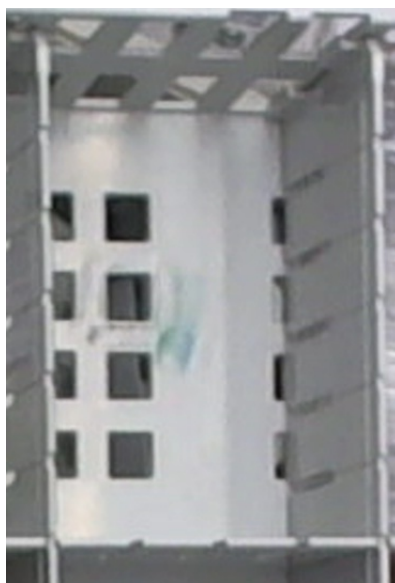


Figure 5.13 Exemple de casier avec un fond en partie quadrillé

Pour faire fonctionner notre méthode avec les fonds semi-quadrillés et semi-lisses, on peut tenter de la modifier pour l'adapter à ce problème. Une idée possible serait d'appliquer la technique de croissance de région pour tenter de couvrir également la partie lisse du tiroir. La section 5.3 décrit comment fonctionne cette technique. Il faut d'abord choisir une graine de départ avant d'effectuer les itérations de croissance de région. Ici, notre méthode pourrait utiliser les carrés trouvés précédemment comme graines puisque nous savons que ces carrés font partie du fond du tiroir. La figure 5.14 illustre le résultat obtenu lorsque l'on combine la détection des carrés avec une croissance de région.

Une fois que la croissance de région est terminée, il devient possible de faire une meilleure estimation de la portion vide du casier. On effectue le même calcul que celui montré plus tôt, mais en incluant l'aire de la région obtenue à celle des carrés. Ce nouveau rapport d'aires devrait normalement indiquer une meilleure estimation de l'état du casier.

5.5.8 Limitations

La méthode de classification de casiers de cette section est expliquée avec des exemples d'images donnant de bons résultats. Il y a cependant certaines limitations qui peuvent apparaître dans d'autres situations. Par exemple, il peut arriver que les items du tiroir soient de couleur semblable à celle du fond du tiroir. Dans ce cas-ci, les carrés sont quand même bien



Figure 5.14 Détection de carrés (en rouge) suivie d'une croissance de région (en rose)

détectés, mais lorsque l'étape de croissance de région est effectuée, la région déborde souvent sur ces items gris. Ceci fait en sorte que l'aire de la région correspondant au fond du tiroir devient trop grande et donne une mauvaise indication de l'état du casier.

Mis à part la qualité des résultats obtenus avec cette méthode, un autre inconvénient important est le temps d'exécution de l'algorithme. Étant donné que cette méthode requiert plusieurs étapes, on peut s'attendre à ce que son exécution soit plus longue. Les étapes les plus longues sont sans doute la détection de lignes par Burns, puis la croissance de région. L'appariement des lignes pour obtenir des carrées ne requiert pas trop de calculs puisque la quantité de lignes à considérer reste relativement petite. On peut aussi s'attendre à ce que le fait d'augmenter la résolution de l'image permette d'obtenir de meilleurs résultats. Par contre, une image contenant plus de pixels fera aussi en sorte que le temps d'exécution sera allongé. Comme les tiroirs du haut du cabinet sont plus près de la caméra que ceux du bas, on peut prévoir que les tiroirs du haut prendront plus de temps à être traités, mais donneront aussi des résultats plus fiables.

5.6 Détection du contenu par segmentation de couleur

5.6.1 Justification

Les méthodes de classification de casier montrées jusqu'à présent sont souvent limitées lorsque les items ressemblent trop au fond du tiroir. Par exemple, des items lisses, comme des boîtes de carton, rendent difficile la classification de casiers par les contours ou par la

croissance de région. La classification par la détection de fonds quadrillés peut aussi être problématique avec des items trop lisses, à cause de la dernière étape de croissance de région. Quant à la soustraction d'arrière-plan, cette méthode fonctionne moins bien avec des items teints d'un gris comme celui du fond de tiroir.

Une solution possible à ces problèmes est de modifier les fonds de tiroir en les colorant d'une couleur particulière. Pour colorer ces fonds, on pourrait le faire en appliquant une peinture sur les tiroirs, mais peindre chaque tiroir demanderait du temps. Une idée plus efficace est d'utiliser des feuilles en plastique colorées et suffisamment minces pour pouvoir être insérées sous les séparateurs des tiroirs. La figure 5.15(a) montre l'allure que peuvent prendre ces feuilles de plastique. Des carrés ont aussi été découpés dans ces feuilles pour être compatibles avec les fonds de type quadrillé.

5.6.2 Choix de la couleur du fond

Le plastique doit être teinté d'une couleur facilement reconnaissable, qui ne sera pas confondue avec d'autres items. On peut penser à certaines couleurs qui sont rarement utilisées sur les emballages de produits. De bons exemples de couleurs sont celles que l'on appelle fluorescentes. En fait, ces couleurs ne sont pas réellement fluorescentes parce qu'elles n'émettent aucune émission lumineuse. Des couleurs semblables sont aussi utilisées avec la technique d'incrustation vidéo. Cette technique consiste à filmer un objet ou une personne devant un fond bleu ou vert pour que ce fond puisse être remplacé par une autre séquence vidéo. On choisit le bleu ou le vert parce que ces teintes sont très peu présentes dans la couleur de la peau d'une personne.

Dans notre cas, en plus de penser au côté fonctionnel du projet, l'entreprise Logi-D nous demandait aussi de faire attention au côté esthétique. Pour que les clients de Logi-D soient intéressés par le projet, la couleur utilisée ne devait pas être trop dérangeante. Par exemple, une couleur verte fluorescente aurait été très pratique d'un point de vue fonctionnel, mais moins intéressant d'un point de vue esthétique. La couleur orange a finalement été choisie parce qu'elle reste suffisamment distinguable tout en respectant les exigences de Logi-D. Le orange choisi est celui qui est montré dans la figure 5.15(a).

5.6.3 Choix d'un espace de couleur

Une fois qu'un choix de couleur a été fixé pour les fonds de tiroirs, l'objectif est de pouvoir segmenter cette couleur dans l'image. Pour faire une segmentation, on utilise une technique semblable à celle utilisée lors de la segmentation des marqueurs dans la section 3.2.2. Comme

avec les marqueurs, un espace de couleur approprié pour faire la segmentation doit être choisi. En effet, le choix d'un bon espace peut beaucoup faciliter la segmentation.

En observant la figure 5.15(a), on remarque que des ombres créées par les séparateurs apparaissent sur le fond du tiroir. Ces ombres peuvent poser problème lors de la segmentation, puisqu'ils viennent assombrir la couleur orange du tiroir. Une façon simple et efficace d'éliminer ces ombres de l'image est de séparer ou de retirer l'information de luminance dans l'image. Certains espaces de couleur, comme l'espace RGB normalisé, permettent justement d'avoir cet effet. Pour passer de l'espace RGB à l'espace RGB normalisé, il suffit de normaliser les trois composantes RGB en divisant chacune d'entre elles par la somme des trois composantes. Les équations 5.10 montrent les calculs à effectuer. On remarque que la somme des trois composantes normalisées sera toujours égale à 1.

$$\begin{aligned} r &= \frac{R}{R + G + B} \\ g &= \frac{G}{R + G + B} \\ b &= \frac{B}{R + G + B} \\ r + g + b &= 1 \end{aligned} \tag{5.10}$$

Cette opération de normalisation revient à retirer l'information de luminosité. Le niveau de luminosité peut être ignoré puisqu'il n'est pas nécessaire à la segmentation du orange. La figure 5.15(b) montre le résultat de cette conversion d'espace de couleur. On peut voir dans cette nouvelle image que les ombres qui étaient causées par les séparateurs ne sont plus présentes. Tous les pixels du fonds apparaissent maintenant comme une seule couleur uniforme, ce qui facilite la segmentation du fond.

5.6.4 Segmentation par seuillage

Avec l'image normalisée, la segmentation peut être effectuée en définissant des seuils pour chaque composante RGB. Pour chacune de ces trois composantes, on définit un seuil minimal et maximal, correspondant aux limites de la couleur orange recherchée. Ces six seuils sont définis lors de l'installation du système, pendant la phase de calibration. Les seuils ne peuvent pas être connus à l'avance parce qu'ils dépendent des conditions d'éclairage à l'emplacement des cabinets. Si des fenêtres sont présentes ou si les lumières sont légèrement teintées, cela affectera la valeur des seuils de segmentation. Pour être plus précis, on peut également définir des seuils d'orange différents pour chacun des tiroirs. En effet, selon la hauteur du tiroir et selon son inclinaison, il peut y avoir des variations dans la teinte du orange vu par la

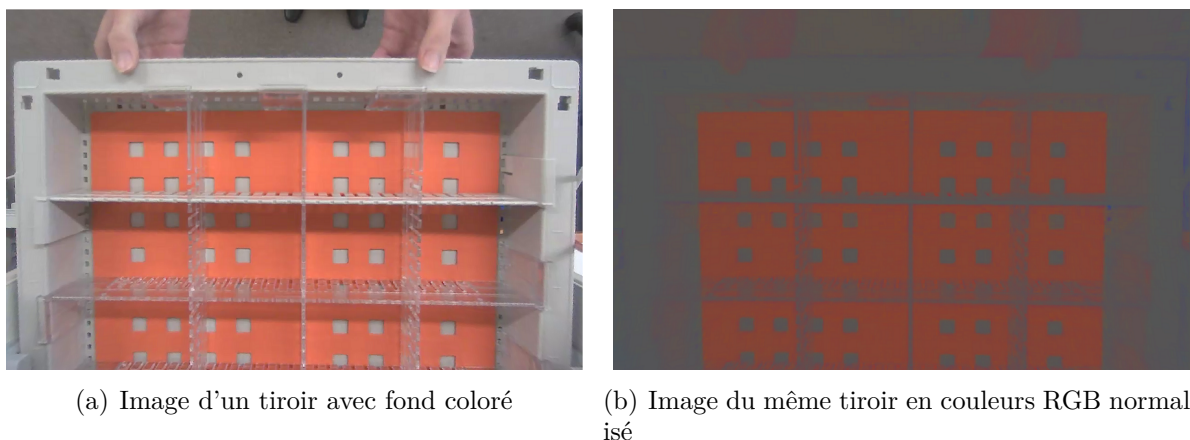


Figure 5.15 Exemple de conversion à l'espace de couleur RGB normalisé

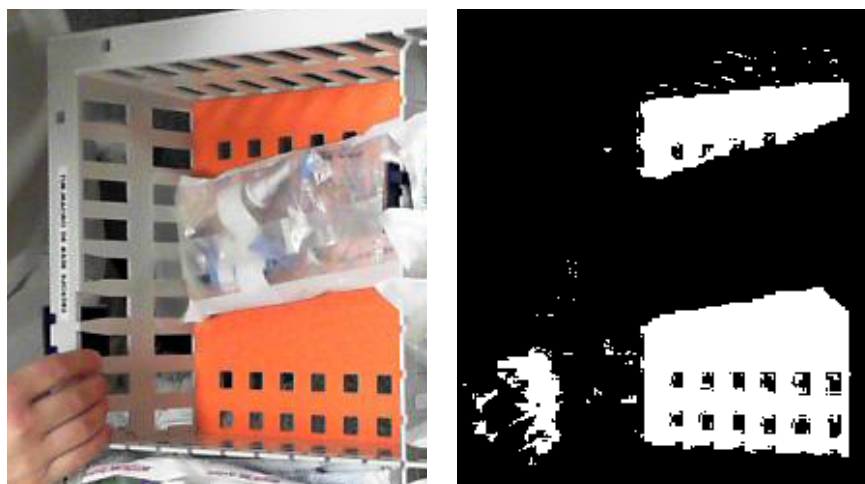
caméra. Étant donné que le plastique utilisé est réfléchissant, il peut aussi arriver qu'un tiroir présente des spécularités causées par les luminaires. En définissant des seuils différents pour chaque tiroir, on définit un intervalle de couleurs orangées qui correspondra mieux au tiroir en question.

La procédure de calibration permet de définir les valeurs des seuils de couleur orange. La procédure que nous avons utilisée demande à l'utilisateur d'ouvrir un tiroir donné, puis de cliquer sur des pixels orange appartenant au fond. Avec les pixels sélectionnés, la procédure enregistre ensuite, pour chaque composante RGB, la valeur minimale et maximale de l'ensemble de pixels sélectionnés. Ces seuils sont finalement enregistrés dans un fichier séparé afin de pouvoir être rechargés en mémoire éventuellement.

Une fois que les seuils sont connus, la classification s'effectue en seuillant chaque pixel indépendamment. Si la couleur d'un pixel se trouve à l'intérieur de l'intervalle défini par nos seuils, on considère qu'il s'agit d'un pixel de fond de tiroir. Les autres pixels se trouvant hors des seuils sont des pixels appartenant à des items d'inventaire. La figure 5.16 illustre un exemple de segmentation possible. L'image binaire 5.16(b) affiche des pixels blancs lorsqu'il s'agit de pixels de fonds.

5.6.5 Résultat de la segmentation

Le résultat de la segmentation à la figure 5.16 donne le résultat attendu. Tous les pixels orange du fond du tiroir ont été reconnus par notre segmentation. Par contre, certains pixels sont segmentés alors que ce ne sont pas des pixels de fond. On remarque d'abord la main en bas à gauche de l'image qui laisse une trace visible dans l'image binaire. On peut comprendre que la main soit en partie segmentée puisqu'elle contient un peu de teintes or-



(a) Image d'un tiroir avec fond orange (b) Résultat de la segmentation du orange

Figure 5.16 Exemple de segmentation d'un fond orange

ange. Ceci serait problématique si la main se trouvait au-dessus d'un item, puisque la main serait alors interprétée comme le fond d'un tiroir. Une façon d'éviter ce problème est de ne pas effectuer de classification lorsqu'une main se trouve à l'intérieur d'un tiroir. Nous avons abordé ce problème à la section 5.1, lorsque nous avons discuté des conditions préalables à la classification.

La figure 5.16(b) nous permet également de voir que quelques pixels qui se trouvent sur les bordures du tiroir ont été segmentés. Ceci s'explique par le fait qu'une partie de la couleur orange du fond est réfléchiée sur les bordures du tiroir. En regardant l'image originale 5.16(a), on voit que les bordures sont légèrement teintées, surtout la bordure en haut de l'image. C'est un phénomène optique qui se nomme radiosité, et que l'on peut observer lorsqu'une surface réfléchit une partie de sa lumière reçue sur une autre surface. La quantité de lumière reste heureusement assez faible pour que l'effet de radiosité soit négligeable sur nos résultats.

5.6.6 Classification à partir de la segmentation

Comme dans les méthodes de classifications précédentes, la dernière étape est de diviser le nombre de pixels détectés par l'aire du casier. Ce calcul nous indique le pourcentage du fond visible pour un casier donné. Il faut noter que même si un casier est vide, son pourcentage n'atteindra jamais 100%. Ceci s'explique par le fait qu'une portion de l'aire du casier est toujours occupée par les séparateurs ou par les trous carrés du tiroir. Par contre, le pourcentage peut atteindre 0% si aucun fond n'est visible.

Pour avoir un meilleur indicateur de l'état du casier, il serait préférable de faire varier ce pourcentage entre 0% et 100%. Pour remédier à problème, on peut diviser ce dernier pourcentage par le pourcentage obtenu lorsque le même casier est complètement vide. L'équation 5.11 décrit le calcul à effectuer pour obtenir ce nouveau pourcentage.

$$\text{Nouveau pourcentage} = 100\% - \frac{\text{pourcentage}_{\text{actuel}}}{\text{pourcentage}_{\text{casier vide}}} \quad (5.11)$$

Le pourcentage du casier vide peut être déterminé lors de la calibration du système. Pour ce faire, on demande à l'utilisateur de vider complètement le casier de ses items. Le système effectue alors la même méthode de segmentation décrite dans cette section, puis calcule le pourcentage de pixels de fond visibles. Ce pourcentage est enregistré, et sera utilisé chaque fois que nous voulons classifier le casier en question. Ce procédé de calibration est effectué pour chacun des casiers, puisque la taille des séparateurs visibles et le nombre de carrés peut varier d'un casier à l'autre.

Le nouveau pourcentage obtenu, qui varie maintenant entre 0% et 100%, est celui qui est utilisé pour faire la classification d'un casier. La classification s'effectue à l'aide d'un seuillage sur la valeur de ce pourcentage. Si le pourcentage se trouve sous un certain seuil choisi, le casier est alors classifié comme étant vide. Dans nos expériences, nous avons utilisé un seuil de 20% pour classifier nos casiers.

5.6.7 Avantages et inconvénients

Par rapport aux autres méthodes de classifications vues, la classification par segmentation est surtout avantageuse pour sa simplicité et sa vitesse d'exécution. Elle demande très peu de calculs et peut presque être effectuée en temps réel, sur un ordinateur suffisamment performant. Le principal inconvénient reste toujours le problème des occlusions, mais ce problème existera, peu importe la méthode de classification utilisée. Dans nos expériences, c'est la méthode de classification par segmentation que nous avons utilisée. Des résultats avec cette méthode seront présentés dans le chapitre 6.

CHAPITRE 6

RÉSULTATS ET DISCUSSION

6.1 Introduction

Dans ce chapitre, nous présentons les résultats obtenus grâce aux différents algorithmes décrits dans les chapitres 3, 4 et 5. Pour être testés, ces algorithmes ont été implémentés avec le langage de programmation Matlab. Ce logiciel a été choisi parce qu'il permet de développer et de prototyper des algorithmes plus facilement et plus rapidement. Le seul inconvénient majeur avec Matlab est que la vitesse d'exécution des algorithmes peut être plus lente. Par contre, la vitesse d'exécution était moins importante dans les premières phases du projet, puisque c'est surtout la qualité des résultats qui nous importaient. De plus, une fois que ces algorithmes fonctionnent, il est toujours possible de les accélérer en les codant dans un autre langage comme C ou C++ avec une librairie de traitement d'images comme OpenCV.

Ensuite, une fois que ces algorithmes étaient implémentés, l'entreprise Logi-D nous a permis d'effectuer des tests grâce à un projet pilote fait chez un de ses clients. L'Hôpital du Sacré-Cœur de Montréal nous a donné la permission d'installer quelques caméras ainsi que tout le matériel nécessaire dans la réserve de sa clinique de chirurgie d'un jour. Sur chacun des 5 cabinets de la réserve, nous avons d'abord installé un support à caméra pour que chaque caméra soit fixée au-dessus de son cabinet. Chacune de ces caméras était reliée à un même ordinateur, placé dans la même pièce, qui s'occupait de traiter les vidéos enregistrées par les caméras. Des marqueurs de couleur ont aussi été apposés sur chacun des tiroirs de la réserve, de façon à permettre la détection et l'identification de tiroir tel que décrit dans les sections 3.2 et 3.3. Nous avons aussi dû vider chacun des tiroirs pour y placer des fonds de tiroirs en plastique coloré comme celui de la figure 5.15(a). Ces fonds étaient utilisés dans le but de permettre la méthode de classification par segmentation de couleur décrite à la section 5.6.

Grâce à ces installations, nous avons pu recueillir de nombreuses séquences vidéos dans lesquelles le personnel effectuait des manipulations d'inventaire de la réserve. En faisant ces installations, nous avons décidé de ne pas modifier le système déjà en place de réapprovisionnement par étiquette RFID décrit à la section 1.2. On demandait aussi au personnel de ne pas modifier ses habitudes de travail et de continuer à utiliser les étiquettes RFID comme il le faisait auparavant. La raison pour laquelle nous voulions garder en place l'ancien système

était de pouvoir comparer ses résultats avec le nouveau système de réapprovisionnement par caméra. Idéalement, il faudrait que les deux systèmes nous indiquent d'effectuer des réapprovisionnements des mêmes items et aux mêmes moments. Nous avons également accès aux données de réapprovisionnement de l'ancien système dans le but de pouvoir les comparer avec le nouveau système. En bref, ce projet pilote nous a permis de tester notre système dans des cas d'utilisations réels que nous n'aurions peut-être pas pensé reproduire en laboratoire.

Dans ce chapitre, nous comparerons dans un premier temps les deux méthodes de détection de tiroir présentées aux sections 3.2 et 3.4, c'est-à-dire la méthode avec marqueurs et celle sans marqueur. Dans un deuxième temps, nous présenterons les résultats de notre système obtenus à l'Hôpital du Sacré-Cœur, en comparaison avec les résultats de l'ancien système déjà en place.

6.2 Méthodes de détection de tiroir

6.2.1 Mise en contexte

Le premier aspect du projet à évaluer est celui de la détection de tiroir. Dans le chapitre 3, nous décrivions deux méthodes pour détecter et identifier les tiroirs. La première méthode de la section 3.2 faisait l'utilisation de marqueurs pour faciliter le repérage des tiroirs. La deuxième méthode de la section 3.4 utilisait plutôt un algorithme de détection de lignes pour détecter les côtés des tiroirs. Dans cette section, nous tentons de comparer ces méthodes afin de déterminer si une des deux méthodes est préférable.

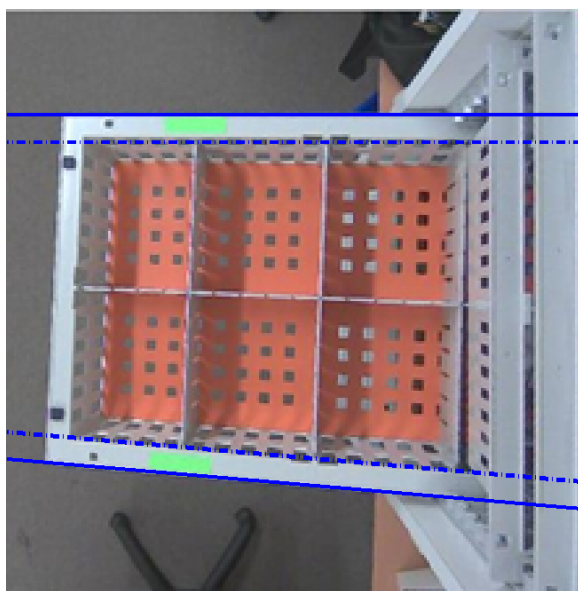
Pour comparer ces algorithmes, nous utilisons une banque de séquences vidéo enregistrées par une caméra d'un cabinet de l'Hôpital du Sacré-Cœur de Montréal. Les vidéos ont été enregistrées par cette caméra sur une période de 16 jours consécutifs, bien que la réserve n'était utilisée que pendant les jours ouvrables. Il y a donc eu 12 jours ouvrables pendant lesquels nous avons pu enregistrer 195 vidéos de manipulations d'inventaire par le personnel. Des marqueurs de couleur avaient été posés sur les tiroirs afin de permettre la détection par la première méthode.

La comparaison des deux méthodes de détection s'effectue à l'aide des vidéos de la banque. Chacune des vidéos de cette banque est traitée deux fois, c'est-à-dire une fois pour chacune des deux méthodes présentées. Les résultats des détections sont ensuite comparés entre eux afin de déterminer si une méthode est plus performante.

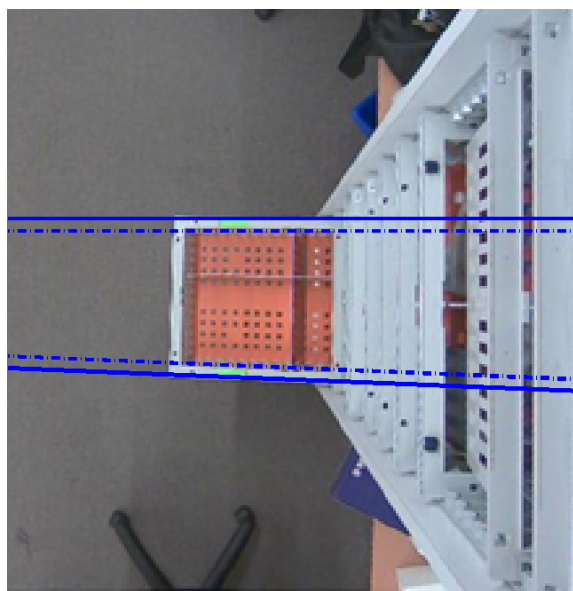
En effectuant quelques traitements vidéo par la deuxième méthode de détection de lignes, nous avons remarqué que cette méthode pouvait parfois être instable. Plutôt que de détecter le côté du tiroir, il pouvait parfois arriver que l'algorithme détecte un séparateur du tiroir ou un autre objet comme un chariot servant à transporter des items. Pour rendre cette méthode

de détection plus robuste dans de telles situations, nous avons choisi de la combiner à celle de la détection de marqueurs.

La combinaison de ces deux algorithmes se fait de la façon suivante. Les lignes sont d'abord détectées tel que décrit dans la section 3.4. Une fois que les lignes sont trouvées, plutôt que de conclure qu'il s'agit d'un tiroir, on effectue une vérification en s'assurant qu'une paire de marqueurs est présente aussi. Cette paire de marqueurs doit coïncider avec la paire de lignes trouvées pour que la détection soit concluante. Pour déterminer si les marqueurs coïncident avec les lignes, on définit des marges de recherche en forme de bandes le long des côtés du tiroir, tel que montré dans la figure 6.1. Dans la figure, les lignes bleues continues correspondent aux lignes qui ont été trouvées par la méthode de détection de la section 3.4. Les lignes bleues pointillées sont déduites à partir des lignes continues afin de former des bandes qui indiquent où rechercher les marqueurs. La largeur de ces bandes doit être proportionnelle à la hauteur du tiroir, puisque plus un tiroir est haut, plus ses marqueurs apparaîtront larges dans l'image. On peut observer cette différence de largeur entre les figures 6.1(a) et 6.1(b). La largeur de ces bandes est calculée par une règle de trois. Il suffit de mesurer au préalable la largeur réelle du tiroir et des marqueurs, puis de calculer la largeur apparente des tiroirs dans l'image pour déduire la largeur que devraient prendre les marqueurs dans l'image.



(a) Bandes pour un tiroir haut



(b) Bandes pour un tiroir bas

Figure 6.1 Exemples de bandes utilisées pour la recherche de marqueurs

Enfin, si aucun marqueur n'est trouvé à l'intérieur de ces bandes, on en conclut qu'aucun tiroir n'est visible dans l'image. Ceci permet d'éviter les fausses détections de tiroir par les

côtés, puisque l'algorithme de détection des lignes peut parfois être sensible au bruit causé par les autres objets.

6.2.2 Résultats et discussion

La table 6.1 montre des exemples de résultats qui sont compilés lors de l'exécution des deux algorithmes de détection de tiroir, pour chacun des vidéos de la banque. À chaque ligne du tableau, on retrouve la date et l'heure de la vidéo en question. La troisième colonne du tableau indique le nombre total de trames contenu dans la vidéo. Ceci permet d'avoir une idée de la durée du vidéo, mais surtout du nombre maximal de détections possibles. Les deux dernières colonnes contiennent les résultats des analyses pour chacun des deux algorithmes. Les résultats de la première méthode, celle utilisant uniquement les marqueurs, sont montrés à l'avant-dernière colonne. Les résultats de la deuxième méthode, celle se servant à la fois des marqueurs et des lignes du tiroir, se trouvent à la dernière colonne du tableau. Pour chacune de ces deux méthodes, on indique le nombre de trames où il y a une détection par la méthode dans chaque vidéo.

Les vidéos sont classées par ordre chronologique. Seulement les cinq premières et les cinq dernières vidéos sont montrées, puisque le tableau complet des 195 vidéos serait trop long pour être présenté ici. On peut aussi remarquer que les vidéos ont surtout été enregistrées durant le jour puisque la réserve et le département étaient fermés la nuit. Enfin, la dernière ligne du tableau contient les totaux pour l'ensemble des trois dernières colonnes.

Tableau 6.1 Exemple de compilation des détections de tiroirs pour une période de 16 jours

Date	Heure	Nombre total de trames	Nombre de trames avec marqueurs détectés	Nombre de trames avec lignes et marqueurs
15 avril 2013	06:30:51	823	529	239
15 avril 2013	06:32:18	268	129	3
15 avril 2013	07:23:54	123	123	4
15 avril 2013	07:24:20	283	102	18
15 avril 2013	07:26:47	144	142	0
...
30 avril 2013	09:04:02	209	69	45
30 avril 2013	09:04:54	591	116	69
30 avril 2013	09:05:25	511	241	145
30 avril 2013	13:20:02	407	136	44
30 avril 2013	13:20:20	391	159	62
Total:		78 902	35 843	10 040

Ces résultats nous permettent d’abord de comparer le nombre de détections de chacune des deux méthodes. On remarque que ce nombre est beaucoup plus élevé avec la première méthode que la deuxième. On pouvait s’attendre à ceci puisque la deuxième méthode détecte non seulement les marqueurs, mais aussi les côtés du tiroir. Cependant, l’écart entre ces deux méthodes est beaucoup plus grand que ce que l’on prévoyait. En effet, la première méthode effectue en moyenne environ 3.6 fois plus de détections que la deuxième : 35 843 détections / 10 040 détections $\cong 3.57$.

On pourrait penser que la première méthode effectue de fausses détections, c’est-à-dire que certaines détections sont faites alors qu’aucun tiroir n’est présent. Par contre, en observant les trames où une détection était faite par la première méthode, nous avons vu qu’il s’agissait presque toujours d’un tiroir. La couleur distincte des marqueurs faisait en sorte qu’elle n’était presque jamais confondue avec la couleur d’un autre objet. L’écart entre les deux méthodes s’explique plutôt par le fait que la deuxième méthode présentait certaines lacunes qui l’empêchaient d’effectuer toutes les détections.

Nous avons essayé de comprendre quelles pouvaient être ces lacunes afin de pouvoir améliorer l’algorithme. Pour cela, nous avons observé les vidéos analysées par la deuxième méthode. Dans ces vidéos, il arrivait, dans 84 des 195 vidéos, que le tiroir soit ouvert dès la première trame, ou que les manipulations soient déjà commencées. Ceci peut s’expliquer par le fait que la caméra utilisée commençait son enregistrement en retard, parce que la détection du mouvement n’avait pas été faite à temps. Pour avoir des enregistrements faits à temps, il aurait fallu mieux ajuster la sensibilité au mouvement de cette caméra ou bien utiliser une autre caméra. Une autre raison pour laquelle un tiroir était déjà ouvert dans la vidéo était que le personnel pouvait parfois négliger de fermer certains tiroirs.

Le fait qu’un tiroir soit déjà ouvert dans la vidéo n’empêchait pas la première méthode de bien fonctionner, puisque les marqueurs restaient toujours visibles. Par contre, comme la deuxième méthode doit effectuer une soustraction d’arrière-plan avant de pouvoir détecter les côtés du tiroir, cette soustraction était alors faussée par la première trame. L’algorithme de soustraction d’arrière-plan considérait que la première trame ne contenait aucun tiroir et qu’elle pouvait donc être utilisée comme une trame d’arrière-plan. Le tiroir était donc soustrait de toutes les trames suivantes de la séquence vidéo, et la détection par les côtés du tiroir devenait difficile.

Cette faiblesse dans l’algorithme de soustraction d’arrière-plan est la principale cause qui explique l’écart entre les deux méthodes de détection du tableau 6.1. D’autres raisons expliquent aussi pourquoi la deuxième méthode ne performe pas aussi bien que la première. En revoyant les vidéos de la banque, nous avons pu remarquer que les côtés du tiroir étaient parfois cachés par le personnel. Par exemple, en effectuant des manipulations ou en se pen-

chant au-dessus d'un tiroir plus bas, le personnel cachait en même temps les côtés du tiroir à détecter. D'autres fois, le personnel n'ouvrait pas suffisamment le tiroir et glissait seulement la main pour aller chercher un item. Dans ces situations d'occlusion, la deuxième méthode ne performe pas aussi bien que la première parce qu'elle requiert que tous les côtés du tiroir soient visibles. Lorsqu'une portion d'un côté de tiroir est masquée, cela est suffisant pour que ce côté ne soit pas détecté. Nous avons pu voir que 11 des 195 vidéos présentaient des situations d'occlusion, que ce soit par les mains, par la tête, ou parce que le tiroir n'était pas tiré complètement. Ce genre de situation est plus difficile à contrôler et la seule façon d'y remédier serait de former le personnel pour éviter des occlusions.

Une dernière situation dans laquelle la deuxième méthode était moins performante est celle des tiroirs inclinés comme dans la figure 3.7. Avec les tiroirs inclinés, les marqueurs sont quand même détectés, mais les côtés ne le sont pas. On comprend pourquoi en regardant l'image d'un tiroir incliné vu de haut comme à la figure 5.16(a). On peut voir que, dans ce cas, les côtés du tiroir ne sont plus droits dans l'image, mais en diagonale. Comme la détection de lignes est programmée pour ne détecter que les lignes droites, les côtés du tiroir incliné ne sont pas détectés. Des tiroirs inclinés étaient présents dans seulement 4 des 195 vidéos de la banque.

Finalement, on peut comparer les deux méthodes en ne considérant que les vidéos où les conditions d'utilisations sont bonnes. Pour cela, on retire tous les vidéos de la banque où des occlusions sont présentes, des tiroirs sont inclinés ou bien des tiroirs sont ouverts avant l'enregistrement. On obtient alors un total de 9 076 détections pour la première méthode et de 6 033 pour la deuxième méthode. On peut voir que la première méthode reste quand même plus performante et effectue environ 1.5 fois plus de détections que la deuxième. Ceci peut s'expliquer par le fait que les marqueurs sont plus souvent visibles que les côtés du tiroir.

En conclusion, à partir de tous ces résultats, on peut recommander de plutôt utiliser la première méthode, parce qu'elle permet de détecter un plus grand nombre de tiroirs en général. La deuxième méthode devrait être améliorée davantage avant de pouvoir être utilisée. C'est pour cette raison que nous avons décidé d'adopter la première méthode de détection de tiroir pour la suite de nos autres analyses.

6.3 Comparaison avec le système “2Bin-iD” de l'Hôpital du Sacré-Cœur

6.3.1 Mise en contexte

Le but de cette section est de comparer les résultats de notre système de détection d'inventaire avec celui déjà en place de l'Hôpital du Sacré-Cœur. Pour ce faire, nous avons installé notre système de caméras dans une réserve de l'hôpital, tel que décrit dans la section 6.1.

L'ancien système d'étiquettes RFID a été laissé en place et le personnel a été avisé de continuer à utiliser ce système comme auparavant. De cette façon, il était possible de recueillir des données provenant de ces deux systèmes et plusieurs séquences vidéos ont pu être enregistrées. Pour effectuer notre analyse, nous avons utilisé les vidéos enregistrées entre le 4 février et le 5 mars 2013 provenant de 2 cabinets.

L'ancien système "2Bin-iD" fonctionne avec des étiquettes RFID tel que décrit à la section 1.2. Lorsqu'un compartiment est vide, le personnel doit prendre l'étiquette de ce compartiment et la poser sur un tableau lecteur de puces RFID. Lorsque la puce est lue par le tableau, une commande de réapprovisionnement est alors envoyée au magasin général de l'hôpital. Enfin, lorsqu'un magasinier s'occupe de faire le réapprovisionnement de la réserve, il doit en même temps replacer les étiquettes du tableau sur les compartiments correspondants. Comme ce système était encore utilisé par le personnel, nous pouvions vérifier si les données de réapprovisionnement concordaient avec celles de notre système.

Avant de pouvoir comparer les résultats de ces deux systèmes, il faut d'abord s'assurer que le format de données de chacun des systèmes soit semblable. Le système "2Bin-iD" génère une donnée chaque fois qu'une étiquette RFID est posée ou retirée du tableau. Cependant, le système de l'Hôpital du Sacré-Cœur est configuré pour ajouter un délai de 30 minutes entre le moment où l'étiquette est déplacée sur le tableau ou sur un compartiment, et le moment où la donnée est générée. Ceci permet d'éviter les erreurs de manipulation, et permet à l'utilisateur de corriger ses erreurs avant que l'information ne soit déjà envoyée au système de réapprovisionnement. Le fait que la commande de réapprovisionnement soit envoyée 30 minutes plus tard ne cause pas de problème, puisque des items sont toujours disponibles dans les casiers secondaires, que nous avons présentés à la section 1.2.

Quant au système par caméra, aucun délai n'a été configuré entre le moment où un casier est détecté comme vide et le moment où la demande de réapprovisionnement est générée. Cette différence entre les deux systèmes doit être considérée afin de pouvoir comparer correctement les données entre elles. Ceci signifie qu'une valeur 30 minutes doit d'abord être soustraite aux heures enregistrées du système "2Bin-iD", avant qu'une comparaison des deux systèmes ne puisse être effectuée. De cette façon, les deux systèmes génèrent des données de façon similaire qui peuvent être comparées entre elles.

Pour chacun de ces deux systèmes, on définit le concept d'évènement comme le fait qu'une demande de réapprovisionnement soit générée ou complétée. Pour le système "2Bin-iD", un évènement se produit donc lorsqu'une étiquette est placée sur le tableau ou retirée du tableau. Quant à notre système, les évènements se produisent lorsque le statut d'un casier change de "plein" à "vide" ou de "vide" à "plein". Ces transitions entre les états "plein" et "vide" se produisent lorsque le pourcentage de détection de fond franchit un certain seuil, tel que décrit

à la section 5.6.6. D’après les expériences que nous avons faites, nous avons déterminé qu’un seuil de 20% permet de faire une bonne classification entre les casiers vides et pleins.

6.3.2 Résultats et discussion

L’analyse des résultats de notre système est effectuée en comparant les événements générés par chacun des deux systèmes. On suppose que les événements générés par le système “2Bin-iD” sont bons et que le personnel a effectué les bonnes manipulations d’étiquettes. En d’autres mots, le système “2Bin-iD” sert de référence afin de déterminer si les événements de notre système sont corrects ou non. Pour chacun des événements générés par le système “2Bin-iD”, notre procédure d’analyse tente de retrouver un événement correspondant de notre système. On considère que deux événements correspondent si leurs dates, leurs heures et leurs types d’événement sont les mêmes. Deux événements sont du même type s’il s’agit d’une étiquette posée sur le tableau et d’un casier dont le statut change de “plein” à “vide”, ou encore, s’il s’agit d’une étiquette retirée du tableau et d’un casier dont le statut change de “vide” à “plein”. Il n’est pas nécessaire que l’heure de deux événements correspondants soit exactement la même puisque le personnel peut parfois prendre un peu de temps avant de déplacer une étiquette RFID. Dans notre analyse, nous permettons que les heures de deux événements soient différentes de 30 minutes. Au-delà de ce seuil, deux événements sont considérés comme étant différents.

Avant de chercher à établir des correspondances d’événements entre les deux systèmes, il peut être pratique d’établir une liste des détections enregistrées pour chacun des produits de l’inventaire. À partir des analyses vidéo faites, il est possible d’établir un historique des pourcentages mesurés pour chaque compartiment à chacune des trames vidéos. On peut ensuite générer des graphiques à partir de ces mesures. La figure 6.2 montre un exemple de graphique qui peut être obtenu pour un produit en particulier.

En abscisse de ce graphique, on retrouve les numéros de trame de séquence vidéo. Notons que, comme les graphiques sont générés à partir d’une seule liste de mesures, ces mesures peuvent provenir de plusieurs séquences vidéos et que ces graphiques n’indiquent pas les moments où il y a des changements de séquences. En ordonnée, on retrouve le pourcentage mesuré par la méthode décrite à la section 5.6.6. En observant l’allure de la courbe, on remarque que le pourcentage mesuré chute vers la trame numéro 50, et qu’il remonte vers la trame numéro 145. À ces deux endroits, le pourcentage franchit notre seuil fixé à 20%. Cela signifie que notre système aura créé un événement de type “casier vide” à la trame 50, et de type “casier plein” à la trame 145.

Les événements du système “2Bin-iD”, quant à eux, sont enregistrés dans une base de données relationnelle SQL, gérée par Logi-D. On peut récupérer ces événements à l’aide

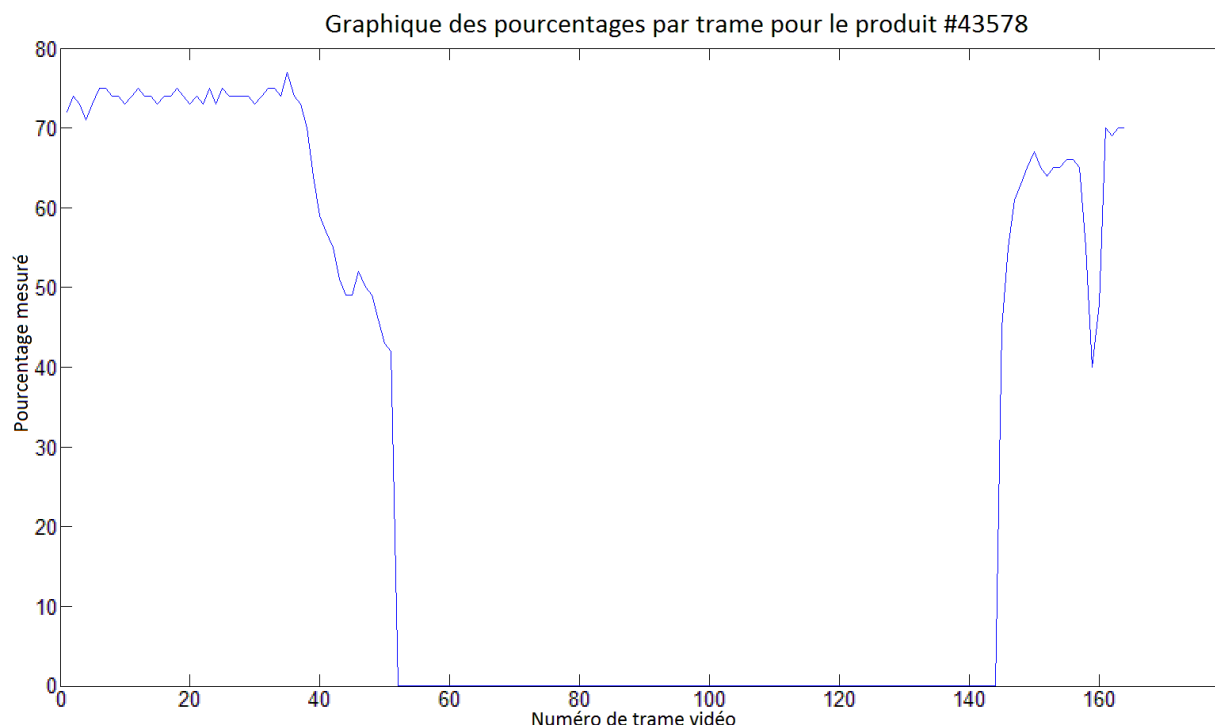


Figure 6.2 Exemple de graphique créé à partir des mesures d'un seul compartiment sur plusieurs trames vidéo

d'une requête SQL appropriée et ainsi obtenir la liste des événements de "2Bin-iD". Une fois que nous avons les listes d'événements provenant des deux systèmes, la prochaine étape est ensuite d'identifier les correspondances d'événements.

Dans notre analyse, nous avons décidé de ne pas considérer les événements provenant de tiroirs inclinés parce que ceux-ci causeraient trop de problèmes de détection pour l'instant. D'abord, la variation de hauteur des marqueurs durant l'ouverture de ces tiroirs rendait l'identification de ces tiroirs plus difficile. Ensuite, comme l'angle de vue de la caméra ne permet pas de voir complètement le fond des tiroirs inclinés, il devenait plus compliqué d'estimer l'état des compartiments pour ces tiroirs. C'est pour ces raisons que notre analyse ne porte que sur les tiroirs droits.

Ainsi, pour la période allant du 4 février au 5 mars 2013, la base de données SQL du système "2Bin-iD" contenait 91 événements pour les tiroirs droits des deux cabinets surveillés par caméra. Pour chacun de ces 91 événements, notre procédure d'analyse a cherché à identifier un événement correspondant parmi la liste d'événements de notre système par caméra. De cette façon, 45 des 91 événements "2Bin-iD" ont pu être associés à un événement correspon-

dant. Ceci signifie que 49.5% des 91 évènements “2Bin-iD” ont également été détectés par notre système de détection d’inventaire. Le tableau 6.2 résume ces données de notre analyse.

Tableau 6.2 Nombre d’évènements correspondants pour la période du 4 février au 5 mars 2013

Total d’évènements “2Bin-iD”	Total d’évènement correspondants	Pourcentage d’évènements détectés par les deux systèmes
91	45	49,5%

Le but de cette analyse, en plus de connaître le niveau de fiabilité de notre système, est aussi de comprendre les causes d’erreurs du système. La seule façon pour nous de connaître ces causes d’erreur était de visionner les vidéos enregistrées. Pour ce faire, nous avons pu récupérer ces vidéos depuis l’ordinateur qui était responsable du traitement vidéo de notre système.

Le visionnement de ces vidéos nous a permis de comprendre pourquoi les 46 des 91 évènements de “2Bin-iD” ne correspondaient à aucun évènement de notre système. Ces 46 conflits entre les deux systèmes sont expliqués par différentes raisons. D’abord, une première explication est que 11 des évènements de “2Bin-iD” avaient été générés sans qu’aucune vidéo soit enregistrée par nos caméras. Il est possible que le seuil de sensibilité de mouvement qui est responsable de déclencher l’enregistrement des vidéos n’était pas suffisamment sensible. Il faut aussi noter que le 28 février en après-midi, une des deux caméras a cessé de fonctionner sans que l’on sache pourquoi. Le fait qu’aucune vidéo n’ait été enregistrée pendant ces 11 évènements explique pourquoi aucun évènement correspondant ne pouvait être généré par notre système.

Ensuite, 21 de ces 46 conflits étaient en fait des erreurs du système “2Bin-iD”. Le système “2Bin-iD”, que l’on considérait comme le système de référence, avait créé ces faux évènements. En effet, il pouvait arriver que le personnel enlève la puce RFID d’un casier alors que ce casier n’était pas complètement vide. Normalement, le personnel doit demander un réapprovisionnement seulement lorsqu’un casier devient vide. Ces demandes de réapprovisionnement hâtives auraient pu être évitées si le personnel avait été mieux avisé de cette consigne.

Enfin, les 14 conflits restants étaient dus à des erreurs de détection de notre système. Les différentes causes de ces 14 erreurs seront expliquées plus en détail dans la suite de cette section. En résumé, le tableau 6.3 énumère les 3 causes d’erreurs pour lesquelles il y a eu 46 conflits entre les deux systèmes. Le tableau indique aussi le nombre d’occurrences de chacune de ces causes, ainsi que leurs taux. Avec les taux d’occurrences, on peut constater que le taux d’erreurs de l’ancien système a été plus élevé que notre taux d’erreurs. En fait, il y a

eu 50% plus d’erreurs de la part de l’ancien système que de la part de notre système. Cette comparaison entre les erreurs des deux systèmes ne tient pas compte des vidéos manquantes.

Tableau 6.3 Causes des conflits entre les deux systèmes

Cause de conflit	Nombre d’occurrences	Taux d’occurrences
Vidéos manquantes	11	12.1%
Erreurs du système “2Bin-iD”	21	23.1%
Erreurs de détection de notre système	14	15.4%
Total	46	50.5%

En revisionnant les vidéos, nous avons aussi pu identifier les différents types d’erreurs de détection de notre système. Voici la description de chacune des causes d’erreurs rencontrées :

- A. **Occlusion du tiroir par la tête de l’utilisateur** : Ce cas se produit lorsque l’utilisateur se penche au-dessus d’un tiroir en bloquant l’angle de vue de la caméra.
- B. **Occlusion d’un marqueur** : Lorsqu’un des deux marqueurs est caché, que ce soit par un item, un bras ou une tête, il devient impossible pour le système de reconnaître le tiroir.
- C. **Tiroir pas assez ou du tout ouvert** : Ceci se produit lorsque l’utilisateur prend un item en glissant sa main dans un tiroir sans l’ouvrir suffisamment pour qu’il soit visible.
- D. **Ombre de l’utilisateur empêchant la détection du tiroir** : Ce cas n’est pas une occlusion, puisque le tiroir est encore visible par la caméra. Par contre, l’ombre de l’utilisateur peut faire en sorte que le marqueur ou le fond du tiroir ne soit pas détecté, puisque leurs couleurs n’avaient pas été calibrées à l’ombre.
- E. **Mauvaise calibration du fond quand le tiroir se ferme** : La façon dont les cabinets ont été conçus fait en sorte que l’angle d’inclinaison des tiroirs, lorsqu’ils sont ouverts, est légèrement en pente vers le bas. Lorsqu’un tiroir est sur le point d’être fermé, son angle d’inclinaison diminue et redevient horizontal. Comme les fonds de plastique sont réfléchissants et que des lumières se trouvaient au plafond, la couleur du fond perçue par la caméra pouvait changer selon l’inclinaison du tiroir qui réfléchissait alors une lumière ou non. De plus, la couleur de fond des tiroirs avait été calibrée en ouvrant complètement les tiroirs et le cas des tiroirs ouverts à moitié n’avait pas été considéré. C’est pour cette raison que la détection du fond pouvait être erronée lorsqu’un tiroir était sur le point d’être fermé ou lorsqu’il était seulement entrouvert.

Le visionnement des vidéos nous a permis de classer chacun des cas d’erreur du système dans une de ces 5 catégories d’erreur. Les nombres d’occurrences pour chaque catégorie ainsi que leurs taux d’occurrences sont montrés dans le tableau de la figure 6.3.

On peut aussi représenter les taux d'occurrences de ce tableau sous la forme d'un diagramme circulaire comme celui de la figure 6.3. Avec ce diagramme, on peut voir plus rapidement quelles sont les causes d'erreurs les plus importantes. Par exemple, on voit que les cas d'occlusions, c'est-à-dire les cas A et B, représentent à eux seuls plus de 70% des erreurs de notre système. Ceci nous permet de voir quelles faiblesses du système devraient être améliorées en priorité afin d'améliorer les résultats.

Causes d'erreurs du système par caméra	Nombres d'occurrences	Taux d'occurrences
A. Occlusion du tiroir par la tête de l'utilisateur	6	42.9%
B. Occlusion d'un marqueur	4	28.6%
C. Tiroir pas assez ou du tout ouvert	2	14.3%
D. Ombre de l'utilisateur empêchant la détection du tiroir	1	7.1%
E. Mauvaise calibration du fond quand le tiroir se ferme	1	7.1%
Total :	14	100.0%

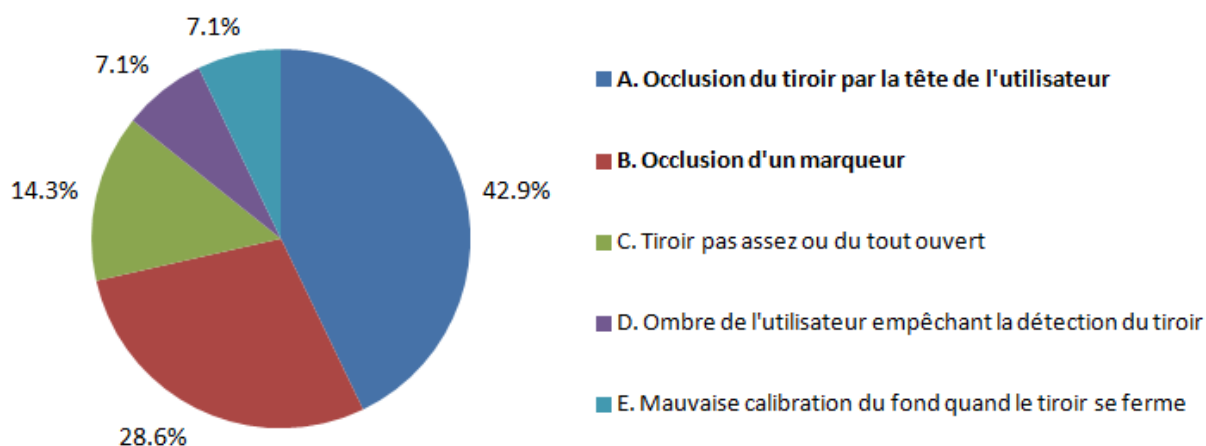


Figure 6.3 Occurrences de chaque type d'erreur du système par caméra

Finalement, suite à ces analyses, il est possible de calculer l'efficacité de notre système en se basant sur les événements réels, plutôt que sur les événements produits par système "2Bin-iD". Ainsi, on peut compter le nombre d'événements réels en soustrayant des événements "2Bin-iD", ceux qui étaient causés par des erreurs du personnel. On peut aussi soustraire les événements où des vidéos étaient manquantes puisque ces erreurs ne sont pas dues à un mauvais traitement vidéo, mais plutôt à un problème spécifique à nos caméras. Le tableau 6.4

résume ce calcul de nombre d'évènements. On obtient donc un total de 59 évènements, qu'un système parfait aurait pu détecter dans des conditions idéales.

Tableau 6.4 Calcul du nombre réel d'évènements filmés

Évènements du système "2Bin-iD" :	+91
Faux évènements de "2Bin-iD" :	-21
Évènements sans enregistrement :	-11
Total d'évènements filmés :	=59

Comme notre système avait détecté 45 de ces 59 évènements, le taux de détection de ces 59 évènements est donc de 76,3%. En d'autres mots, si notre système n'avait eu aucun problème d'enregistrement de vidéos, environ 25% des évènements réels n'auraient pas été détectés à cause des différentes erreurs de détection (A, B, C, D ou E) que nous avons énumérées plus haut.

6.3.3 Comparaison de l'efficacité des deux systèmes

Nous savons maintenant que l'ancien système "2Bin-iD" ne peut pas vraiment servir de référence parce que ce système générerait lui aussi des erreurs. Pour pouvoir comparer ce système avec le nôtre, il pourrait être intéressant de calculer son efficacité. Pour ce faire, on effectue un calcul semblable à celui fait précédemment.

On considère d'abord les 91 évènements enregistrés dans la base de données SQL. Parmi ces 91 évènements, 11 de ces évènements étaient sans enregistrement vidéo. Comme il est impossible de savoir, sans les vidéos, si ces 11 évènements étaient bons ou non, nous ne les considérons pas dans nos calculs. Il reste donc un total de 80 évènements qui se trouvent dans la base de données et qui ont également été enregistrés. Le visionnement des vidéos de ces 80 évènements nous a permis de constater que 21 de ces évènements étaient des erreurs du système "2Bin-iD". En soustrayant 21 évènements aux 80 évènements, on obtient donc 59 évènements comme étant des bonnes détections du système "2Bin-iD". Ceci signifie que le système a pu effectuer 59 bonnes détections parmi 80 de ses détections. Ce rapport nous donne une efficacité de 73,8%.

Finalement, on obtient une efficacité semblable pour les deux systèmes puisque nous avons estimé l'efficacité de notre système à 76,3%. Notre système semble même être légèrement plus efficace d'environ 2,5%. Si deux systèmes ont des efficacités semblables, le choix d'un système pourrait alors se faire en fonction des coûts de chacun. Dans le calcul de ces coûts, il faudrait aussi inclure les coûts associés au remplacement du système déjà en place.

CHAPITRE 7

CONCLUSION

L'objectif de ce travail était d'améliorer la chaîne de réapprovisionnement des produits médicaux dans les hôpitaux. En effet, les systèmes actuels de gestion d'inventaire dans les hôpitaux comportent certaines faiblesses. Par exemple, la plupart des systèmes requièrent que les produits épuisés soient commandés manuellement par le personnel infirmier. Par contre, ces commandes de réapprovisionnement peuvent facilement être oubliées par le personnel, surtout dans le cas d'une situation d'urgence. De plus, des erreurs de commandes peuvent également survenir de la part du personnel. Pour pallier ces lacunes, nous avons proposé de développer un système de vision artificielle par caméra, qui permettrait de repérer les produits manquants à l'inventaire.

7.1 Synthèse des travaux

La revue de littérature nous a d'abord permis de constater que seulement quelques recherches portaient sur la détection du niveau d'inventaire. De plus, la plupart de ces recherches ne concernaient que les commerces au détail. Nous avons donc dû élaborer nous même notre propre système, en nous inspirant de différents algorithmes de traitement d'image. Par exemple, nous avons utilisé des algorithmes de soustraction d'arrière-plan pour la détection du mouvement, des algorithmes de segmentation pour la détection de marqueurs ou des fonds de couleur, ou encore, la transformée de Hough pour détecter les côtés des tiroirs. Le problème de détection d'inventaire peut être divisé en trois sous-problèmes à résoudre indépendamment : la détection et l'identification du tiroir, la détection et la localisation des compartiments, puis la détection des contenus des compartiments.

Le premier sous-problème, celui de la détection et l'identification du tiroir, a été traité au chapitre 3. Dans ce chapitre, nous avons présenté deux approches pour résoudre ce problème. La première approche faisait l'utilisation de marqueurs de couleurs posés sur les tiroirs. Ces marqueurs étaient ensuite détectés par une technique de segmentation. La deuxième approche utilisait plutôt un algorithme de détection de ligne pour détecter les côtés du tiroir.

Le deuxième sous-problème, celui de la détection et la localisation des compartiments, a été abordé au chapitre 4. Encore une fois, deux approches différentes ont été présentées dans ce chapitre. La première était de faire la détection des séparateurs par l'algorithme de

transformée de Hough. La deuxième approche était d'enregistrer, au préalable, les positions des séparateurs pour chacun des tiroirs.

Enfin, le dernier sous-problème, celui de la détection des contenus des compartiments a été analysé au chapitre 5. Selon le type de fond de tiroir utilisé, différentes approches sont possibles. Le fond de type lisse permet les approches de classification par contours, par croissance de région, ou bien par soustraction d'arrière-plan. Quant au fond de type quadrillé, il est possible d'effectuer une classification par une détection des quadrillés. De plus, si le fond est coloré, il est alors possible d'appliquer une segmentation du fond par la couleur.

Une fois que chacun de ces trois sous-problèmes ont été résolus séparément, nous avons pu élaborer un système complet programmé avec le langage de Matlab. Pour chacun de ces sous-problèmes, nous avons choisi l'approche qui nous semblait la plus fiable :

1. La détection des tiroirs se faisait à l'aide de marqueurs.
2. Les positions des séparateurs étaient enregistrées.
3. La détection des contenus des compartiments se faisait par segmentation du fond coloré.

Après avoir été testé en laboratoire, ce système a pu être installé à l'Hôpital du Sacré-Cœur de Montréal en tant que projet pilote. Ceci nous a permis d'évaluer la performance et la fiabilité du système dans des conditions d'utilisation réelle durant plusieurs jours. Ainsi, parmi les 59 événements qui auraient pu être détectés par notre système, 45 événements ont été détectés, c'est-à-dire 76,3% des événements. Les causes d'erreurs du système ont aussi pu être identifiées et comptabilisées. De plus, le visionnement des vidéos enregistrées par notre système nous a permis d'identifier 21 faux événements de la part de l'ancien système. Ces événements n'auraient pas dû être générés par l'ancien système, alors que notre système a réagi correctement en ne générant aucun de ces événements.

En conclusion, nous avons pu développer un système fonctionnel de détection d'inventaire. Par contre, certaines conditions d'utilisation doivent être remplies afin d'optimiser la performance du système. Ces conditions d'utilisations sont détaillées dans la section suivante.

7.2 Limitations de la solution proposée

En effectuant nos tests à l'Hôpital du Sacré-Cœur de Montréal, nous avons pu constater certaines limitations de notre système. Ces limitations ont déjà été présentées en majeure partie dans la section 6.3, lorsque nous avons analysé les causes d'erreurs du système.

Premièrement, la principale limitation du système est que les situations d'occlusions rendent la détection d'inventaire très difficile. Ce problème avait aussi été remarqué par Mustafa et Sethi (2005) lors de la détection d'un tiroir de caisse enregistreuse. Cependant, dans notre cas, les occlusions étaient encore plus fréquentes à cause des tiroirs plus bas. Ces tiroirs

incitaient le personnel à se pencher au-dessus des items de l’inventaire, en créant ainsi des occlusions. En effet, près de 70% des erreurs de notre système étaient causées par des occlusions.

Une deuxième limitation se présente lorsqu’un utilisateur prend un item d’un tiroir sans l’ouvrir suffisamment. Ceci ne permet pas à la caméra de pouvoir capturer le contenu du tiroir et empêche donc la détection de l’inventaire. Dans nos tests à l’hôpital, cette limitation représentait environ 15% des erreurs du système.

Enfin, les autres 15% d’erreurs du système étaient dues à des changements de luminosité non prévus. Par exemple, des spécularités provenant des lumières, ou bien les ombres des utilisateurs faisaient en sorte que les couleurs des marqueurs ou des fonds n’étaient plus reconnues. Cette limitation pourrait cependant être corrigée par une meilleure calibration des couleurs, lors de l’installation du système.

En somme, les limitations du système sont surtout reliées aux conditions d’utilisation du système. La réussite du système dépend beaucoup d’un facteur humain, un aspect qui doit absolument être considéré dans une version future du système. Pour pallier ces limitations, certaines améliorations pourraient être faites dans le futur. Ces suggestions d’améliorations sont décrites dans la prochaine section.

7.3 Améliorations futures

Des améliorations pourraient sans doute être apportées à notre système. Nous décrirons ici celles qui nous semblent les plus évidentes.

D’abord, des premières améliorations pourraient facilement être faites en modifiant légèrement les cabinets. Nous avons présenté une limitation due aux spécularités présentes sur les marqueurs et sur les fonds des tiroirs. Ces spécularités pourraient sûrement être évitées en utilisant un type de plastique plus mat. En effet, le type de matériaux utilisé influence beaucoup la quantité de lumière qui est réfléchi. Une deuxième amélioration aux cabinets concerne le choix de couleur utilisée pour les marqueurs et les fonds des tiroirs. Dans nos expériences, nous avons choisi d’utiliser une couleur orange surtout pour satisfaire aux critères esthétiques de la compagnie Logi-D. Par contre, nous avons remarqué que cette couleur pouvait parfois être confondue avec la couleur de la peau d’une personne. Pour éviter cette confusion, il aurait peut-être été mieux d’utiliser une couleur plus distincte comme un vert fluorescent par exemple. En effet, la couleur verte est beaucoup plus éloignée de la couleur peau que l’est la couleur orange.

Deuxièmement, comme la plupart des erreurs du système étaient dues à des occlusions, une amélioration à ce niveau réduirait beaucoup le nombre d’erreurs. Par contre, même avec

les meilleurs algorithmes de traitement d'image concevables, il serait très difficile de déduire le contenu d'un tiroir masqué. La seule solution à ce problème semble être de demander la collaboration des utilisateurs afin d'éviter les occlusions. Par contre, un utilisateur ne peut pas toujours savoir s'il cache l'inventaire ou non. Une solution possible serait que notre système lui fournisse une information en retour, à la suite d'une tentative de détection. Cette rétroaction indiquerait si le système a pu détecter correctement le contenu d'un tiroir ou non. La rétroaction pourrait être envoyée sous la forme d'un signal sonore ou lumineux comme avec une diode électroluminescente (DEL). De cette façon, le personnel infirmier n'aurait qu'à attendre le signal d'une bonne détection avant de refermer un tiroir.

L'ajout d'une forme de rétroaction à notre système impliquerait cependant certaines modifications. Présentement, les traitements de vidéos sont effectués de manière différée. Pour permettre à l'utilisateur de savoir immédiatement le résultat de la détection, il faudrait que les vidéos soient traitées en temps réel. Initialement, notre système n'avait pas été conçu pour effectuer ses traitements en temps réel. Il faudrait donc améliorer sa vitesse de traitement pour la rendre plus rapide. Dans ce cas, il se pourrait que notre système doive être reprogrammé dans un autre langage de programmation que celui de Matlab, parce que l'environnement de Matlab n'offre pas toujours la meilleure performance. En effet, certaines étapes du traitement sont plus longues à effectuer, comme l'étape de correction de la distorsion de la section 3.1. Peut-être même que des accélérations matérielles pourraient être envisagées pour certaines opérations.

Enfin, une dernière amélioration envisageable serait d'utiliser les informations provenant d'une image de profondeur afin de détecter le contenu des compartiments. Ce genre d'image permet d'avoir, pour chacun des pixels, une valeur de profondeur qui indique la distance entre ce point et la caméra. Si nous savons que le fond d'un tiroir se trouve à une distance donnée, nous pourrions déduire que les pixels d'un compartiment qui se trouve plus près de cette distance sont des pixels de produits. On effectuerait donc une segmentation basée sur l'information de profondeur plutôt que sur les couleurs. Il serait même possible de combiner ces deux informations, celle de profondeur et celle des couleurs, afin d'effectuer une meilleure segmentation.

Il existe plusieurs techniques permettant d'obtenir des images de profondeur. Par exemple, la caméra Kinect utilise une lumière infrarouge appelée "lumière structurée". Selon la déformation de cette lumière, la profondeur peut être déduite à l'aide d'une seule caméra. Il existe aussi les systèmes à deux caméras. Ces systèmes stéréo permettent de calculer la profondeur d'un point donné lorsque ce point est identifié dans chacune des deux images. Par contre, lorsque la surface est lisse et ne présente aucun point caractéristique, il devient difficile de faire correspondre des points entre les deux images. Dans ce cas, il est possible de

projeter une texture sur la surface comme l'on fait Sturm *et al.* (2010) dans leur système de vision stéréo.

Un autre avantage avec les images de profondeur est que certaines occlusions pourraient être facilement détectées. Par exemple, si une tête se trouve au-dessus d'un tiroir, la profondeur des pixels correspondant à la tête sera beaucoup plus petite que la profondeur du tiroir. Dans ce cas, le système pourrait en déduire qu'un autre objet se trouve au-dessus du tiroir.

Nous avons effectué quelques essais avec la caméra Kinect de Microsoft. Le seul problème avec cette caméra est que sa résolution de profondeur n'est pas suffisante pour les tiroirs en bas des cabinets. À cette distance, la précision de l'information de profondeur devient trop faible pour effectuer une bonne segmentation du fond du tiroir. Peut-être qu'un autre type de caméra plus performant nous aurait permis d'obtenir une meilleure résolution.

En conclusion, nous pouvons voir que plusieurs améliorations seraient possibles à notre système. Si des travaux sont réalisés sur ce projet dans le futur, il serait possible d'expérimenter avec ces différentes idées.

RÉFÉRENCES

- BURNS, J. B., HANSON, A. R. et RISEMAN, E. M. (1986). Extracting straight lines. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8, 425–455.
- CANNY, J. (1986). A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8, 679–698.
- DUDA, R. O. et HART, P. E. (1972). Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15, 11–15.
- FÄRNSTRÖM, F., JOHANSSON, B. et ÅSTRÖM, K. (2002). Computer vision for determination of fridge contents. *Proceedings SSAB\ '02, Symposium on Image Analysis*. Swedish Society for Automated Image Analysis, 45–48.
- FISCHLER, M. A. et BOLLES, R. C. (1981). Random sample consensus : a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24, 381–395.
- FRANCIS, J., DROLIA, U., MANKODIYA, K., MARTINS, R., GANDHI, R. et NARASIMHAN, P. (2013). Metabot : Automated and dynamically schedulable robotic behaviors in retail environments. *Robotic and Sensors Environments (ROSE), 2013 IEEE International Symposium on*. IEEE, 148–153.
- GAO, D.-S. et ZHOU, J. (2000). Car license plates detection from complex scene. *Signal Processing Proceedings, 2000. WCCC-ICSP 2000. 5th International Conference on*. IEEE, vol. 2, 1409–1414.
- GONZALEZ, R. C., WOODS, R. E. et EDDINS, S. L. (2003). *Digital Image Processing Using MATLAB*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.
- HANNAN, M., AREBEY, M., BEGUM, R. et BASRI, H. (2011). Gray level aura matrix : An image processing approach for waste bin level detection. *Sustainable Technologies (WCST), 2011 World Congress on*. IEEE, 77–82.
- HANNAN, M., AREBEY, M., BEGUM, R. A. et BASRI, H. (2012). An automated solid waste bin level detection system using a gray level aura matrix. *Waste management*, 32, 2229–2238.
- HOUGH, P. (1962). Method and Means for Recognizing Complex Patterns. Brevet américain 3 069 654.
- IOANNOU, D. et DUGAN, E. (1996). Parallelogram detection in a digital image with the use of the hough transform. *Proceedings of the 13th International Conference on Pattern Recognition*. IEEE Computer Society, Washington, DC, USA, vol. 2 de *ICPR '96*, 532–536.

- JONES, M. J. et REHG, J. M. (2002). Statistical color models with application to skin detection. *International Journal of Computer Vision*, 46, 81–96.
- JUNG, C. R. et SCHRAMM, R. (2004). Rectangle detection based on a windowed hough transform. *Computer Graphics and Image Processing, 2004. Proceedings. 17th Brazilian Symposium on*. IEEE, 113–120.
- LANDRY, S. et BEAULIEU, M. (2010). Achieving lean healthcare by combining the two-bin kanban replenishment system with rfid technology. *International Journal of Health Management and Information*, 1, 85–98.
- LO, R.-C. et TSAI, W.-H. (1995). Gray-scale hough transform for thick line detection in gray-scale images. *Pattern Recognition*, 28, 647 – 661.
- LOWE, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60, 91–110.
- MANKODIYA, K., GANDHI, R. et NARASIMHAN, P. (2012). Challenges and opportunities for embedded computing in retail environments. *Sensor Systems and Software*, Springer. 121–136.
- MCKENNA, S. J., JABRI, S., DURIC, Z., WECHSLER, H. et ROSENFELD, A. (2000). Tracking groups of people. *Computer Vision and Image Understanding*, 80, 42–56.
- MUSTAFA, A. et SETHI, I. (2005). Detecting retail events using moving edges. *Advanced Video and Signal Based Surveillance, 2005. AVSS 2005. IEEE Conference on*. IEEE, 626–631.
- NELSON, R. (1994). Finding line segments by stick growing. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 16, 519–523.
- NGUYEN, T. T., DAI PHAM, X. et JEON, J. W. (2009). Rectangular object tracking based on standard hough transform. *Robotics and Biomimetics, 2008. ROBIO 2008. IEEE International Conference on*. IEEE, 2098–2103.
- OPALACH, A., FANO, A., LINAKER, F. et GROENEVELT, R. B. R. (2012). Planogram extraction based on image processing. US Patent 8,189,855.
- QIN, X. et YANG, Y.-H. (2005). Basic gray level aura matrices : theory and its application to texture synthesis. *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*. IEEE, vol. 1, 128–135.
- ROSEEN, R. (2000). Device and procedure for information on the contents in a cabinet. Patent EP1030521 A1.
- SCHWARTZ, R. (2007). Retail store method and system. US Patent 7,168,618.

- STURM, J., KONOLIGE, K., STACHNISS, C. et BURGARD, W. (2010). Vision-based detection for learning articulation models of cabinet doors and drawers in household environments. *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 362–368.
- TAO, W.-B., TIAN, J.-W. et JIAN, L. (2002). A new approach to extract rectangular building from aerial urban images. *Signal Processing, 2002 6th International Conference on*. IEEE, vol. 1, 143–146.
- VEZHNEVETS, V., SAZONOV, V. et ANDREEVA, A. (2003). A survey on pixel-based skin color detection techniques. *Proc. Graphicon*. Moscow, Russia, vol. 3, 85–92.
- WOLBERG, G. (1990). *Digital image warping*. IEEE Computer Society Press monograph. IEEE Computer Society Press, Los Alamitos, CA, USA.